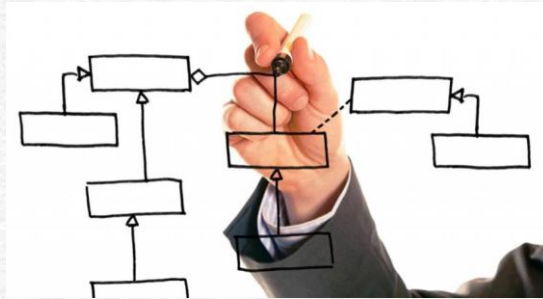


Вступ в програмну інженерію

Основи UML



1

UML — це мова

Універсальна мова моделювання (Unified Modelling Language або UML) — це мова позначень або побудови діаграм, призначена для визначення, візуалізації і документування моделей зорієнтованих на об'єкти систем програмного забезпечення. UML не є методом розробки ПЗ. Розробкою UML керує Object Management Group (OMG). Ця мова є загальноприйнятим стандартом графічного опису програмного забезпечення.

2

UML — це мова

Автори — Гради Буч, Ивар Якобсон и Джеймс Рамбо



Grady Booch
Греді Буч



James Rumbaugh
Джеймс Рамбо



Ivar Jacobson
Айвар Якобсон

3

Призначення UML

Одним з ключових етапів розробки є визначення того, яким вимогам має задовольняти система. В результаті з'являється формальний або неформальний документ (артефакт): постановка завдання, вимоги, технічне завдання, зовнішні специфікації та ін.

Аналогічні за призначенням артефакти з'являються і на інших етапах розробки: функціональні специфікації, архітектура програми та ін. Всі такі артефакти називають *специфікаціями*.

Специфікація - це декларативний опис того, як щось влаштовано або працює.

4

Призначення UML

UML призначений не тільки для *опису абстрактних моделей*, але і для *безпосереднього маніпулювання артефактами*, наприклад, програмним кодом (автоматична генерація програмного коду).

Ще одне призначення UML - документування.

5

Стандарт UML

UML графічна мова, тому правила запису (малювання) моделей є нотацією, а не синтаксисом.

Типи елементів нотації :

- ☛ фігури;
- ☛ лінії;
- ☛ значки;
- ☛ тексти.

6

Інструменти UML

Інструменти	Назва
Ліцензовані	Microsoft Visio IBM Rational Rose
Open source	StarUML ArgoUML Violet UML Editor Astar Community 6.1 BOUML UMLGraph Dia
Плагіни до IDE	Visual Paradigm SDE for Visual Studio NetBeans IDE UML Eclipse UML2 Tools
Онлайн	WebSequenceDiagrams yUML Umbrello UML Modeller Software Ideas Modeler Gliffy

7

Модель UML

Модель UML - це кінцева безліч сутностей і відношень між ними.

Модель UML, це граф (точніше, навантажений мульти-псевдо-гіперорграф), в якому вершини і ребра навантажені додатковою інформацією та можуть мати складну внутрішню структуру.

Вершини цього графа називаються *сутностями*, а ребра - *відносинами*.

8

Сутності

Сутності в UML можна поділити на чотири групи:

- структурні;
- поведінкові;
- групуючи;
- анотуючи.

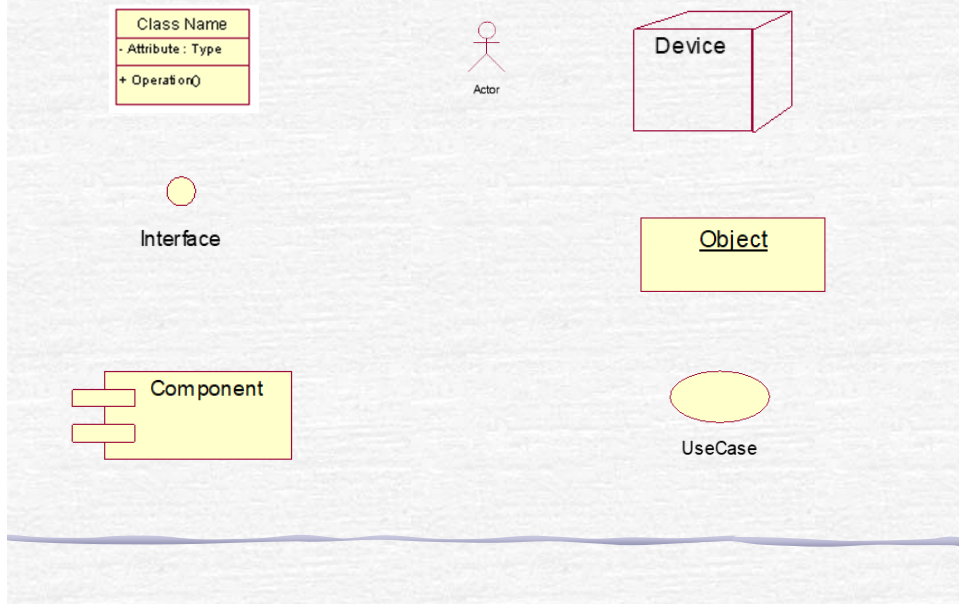
9

Структурні сутності

- *Клас* - опис безлічі об'єктів із загальними атрибутами і операціями.
- *Інтерфейс* - безліч операцій, яке визначає набір послуг (службу), що надаються класом або компонентом.
- *Дійова особа* - сутність, яка перебуває поза модельованої системи і безпосередньо взаємодіє з нею.
- *Варіант використання* - опис послідовності вироблених системою дій, що доставляє значимий для деякої дійової особи результат.
- *Компонент* - фізично замінний артефакт, який реалізує деякий набір інтерфейсів.
- *Вузол* - фізичний обчислювальний ресурс.

10

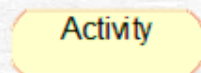
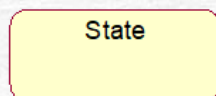
Структурні сутності



11

Структурні сутності

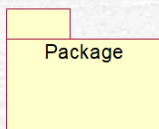
Стан - період в життєвому циклі об'єкта, в якому об'єкт задовольняє деякому умовою, виконує діяльність або очікує події.
Діяльність - стан, в якому виконується робота, а не просто пасивно очікується настання події.



12

Групуючи сутності

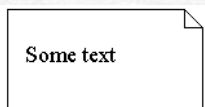
Пакет — група елементів моделі (в том числі пакетів).



13

Анотуючи сутності

Анотації - це сутності, які в UML вживають для представлення пояснень і коментарів. Єдиним типом анотуючої сутності є примітка (note). Примітка з'єднується пунктирною лінією із сутністю, якої вона стосується:



14

Типи зв'язків

В UML використовуються чотири основних типи відносин:

- залежність;
- асоціація;
- узагальнення;
- реалізація.

15

Типи зв'язків

Залежність - це найбільш загальний тип відносини між двома сутностями. Ставлення залежності вказує на те, що зміна незалежної сутності якимось чином впливає на залежну сутність. Графічно відношення залежності зображується у вигляді пунктирною стрілки, спрямованої від незалежної сутності до залежної.

Асоціація - це найбільш часто використовуваний тип відносини між сутностями. Ставлення асоціації має місце, якщо одна сутність безпосередньо пов'язана з іншою (або з іншими - асоціація може бути не тільки бінарною). Графічно асоціація зображується у вигляді суцільної лінії з різними доповненнями, що з'єднує пов'язані сутності.

16

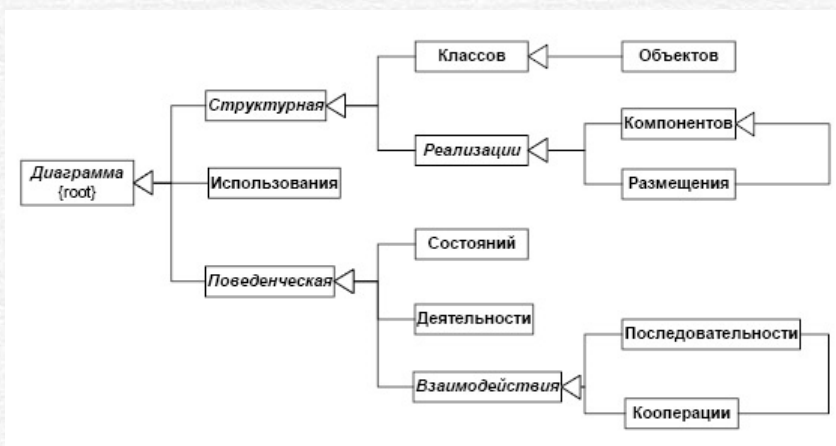
Типи зв'язків

Узагальнення - це відношення між двома сутностями, одна з яких є приватним випадком іншої. Графічно узагальнення зображується у вигляді суцільної стрілки з трикутником на кінці, спрямованої від часткового до загального.

Реалізація - вказує, що одна сутність є реалізацією іншої. Наприклад, клас є реалізацією інтерфейсу. Графічно реалізація зображується у вигляді пунктирною стрілки з трикутником на кінці, спрямованої до реалізованої сутності.

17

Ієрархія діаграм UML



18

Варіанти використання (use cases)

- Діаграми варіантів використання описують поведінку системи з точки зору користувача.
- Вони дозволяють визначити межі системи, а також відношення між системою та зовнішнім середовищем.
- Діаграми варіантів використання використовують для візуалізації вимог до системи.

19

Варіанти використання (use cases)

- Дійові особи:
 - Користувачі системи
 - Інші системи
- Актор не є конкретною людиною, це набір ролей в системі.

20

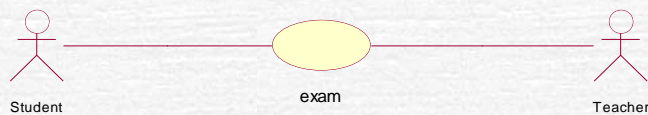
Варіанти використання (use cases)

- Actor – роль, яку грає персона або інша зовнішня сутність під час взаємодії з системою.
- Use case – (варіант використання, або прецедент) відповідає за конкретний спосіб взаємодії з системою. Варіанти використання відображають функціональність системи. описує, **що** робить система, але не вказує **як**.



21

Приклад: Екзамен

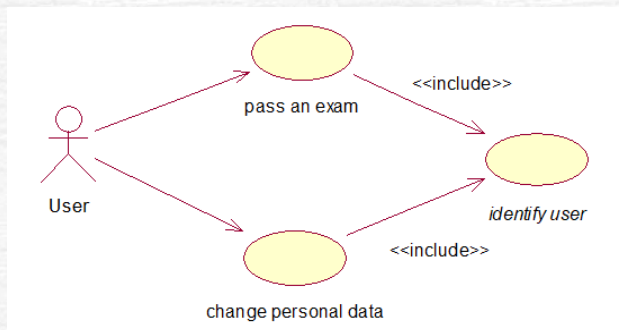


[Teacher розуміє екзамен у Student]

22

Stereotype: <<include>>

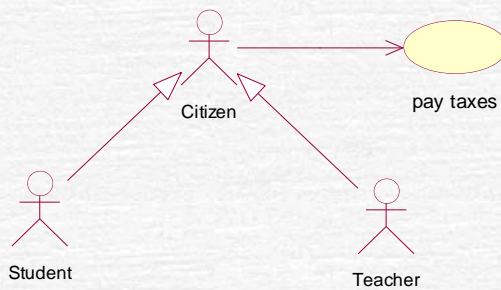
- ✓ Stereotype: <<include>>
- ✓ Різні use-cases можуть мати загальні частини
- ✓ Абстрактний use-case не активується акторами



23

Успадкування в Use Case

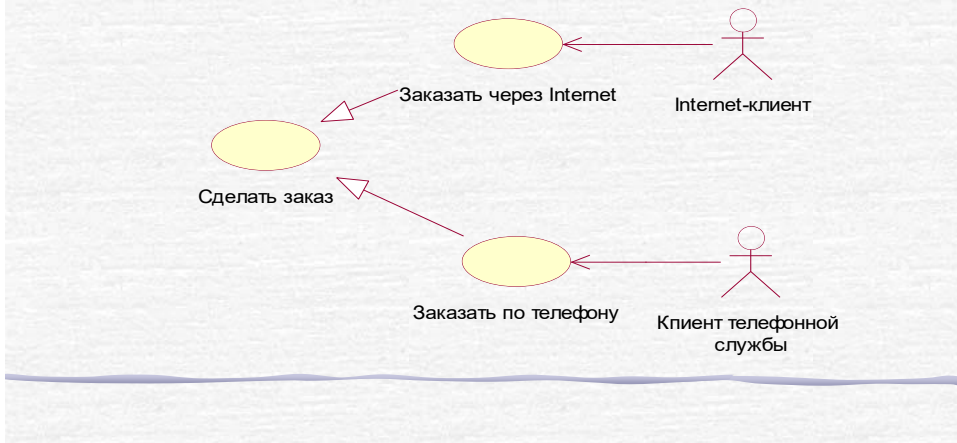
Різні актори можуть грати одну і ту ж загальну роль в деякому use-case



24

Успадкування в Use Case

- Іноді два use-case-а можуть мати деяку загальну послідовність дій.

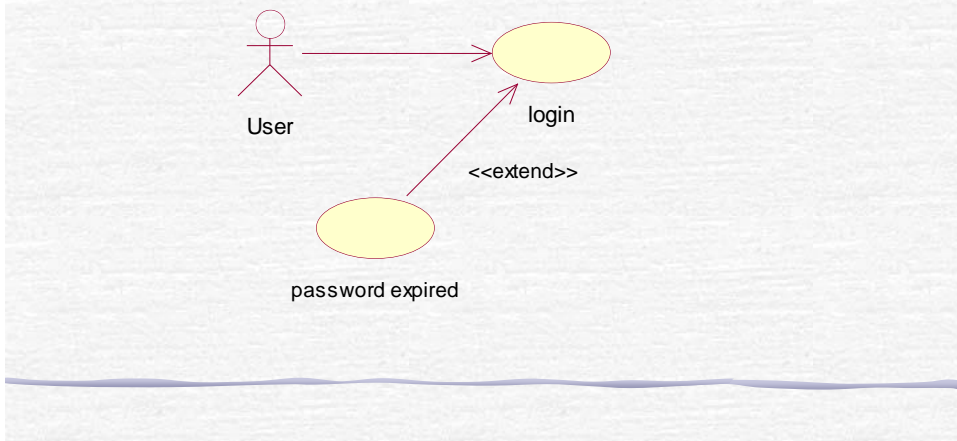


25

Stereotype: <<extend>>

Stereotype: <<extend>>

Деякі use cases можуть викликатися в контексті інших тільки за деяких умов



26

Документування use-cases

- ☞ Ім'я
- ☞ Головна ціль (Goal in context)
- ☞ Головні актори (Primary actor(s))
- ☞ Передумова (Precondition)
- ☞ Умова початку дій (Trigger condition)
- ☞ Сценарії
 - Основний сценарій (main success scenario)
 - Альтернативний сценарій №1 (alternative scenario)
 - ...

27

Документування use-cases

ADM_UC_0.1	Создать учетную запись
Description	Создание учетной записи пользователя Администратором
Actor	Администратор
Pre-Conditions	1. Администратор залогинился в систему 2. У Администратора достаточно прав для создания учетной записи
Main Flow	1. Администратор вызывает функцию 2. Система предлагает ввести информацию про учетную запись 3. Администратор заполняет необходимые данные 4. Система проверяет корректность введенной информации 5. Система создает учетную запись 6. Система уведомляет пользователя (для кого создавалась учетная запись) про создание учетной записи 7. Система уведомляет Администратора про успешность создания учетной записи
Alternative Flow	1.а Администратор отказывается от выполнения операции 1. Система отображает предупреждение 2. Если администратор подтверждает свои действия 2.1 Система заканчивает выполнение основного сценария 3. Если Администратор не подтверждает свои действия 3.1 Система возвращается на Шаг 3 основного сценария 4. а Информация была введена некорректно 4.1 Система уведомляет Администратора про наличие ошибок 4.2 Система возвращается на Шаг 3 основного сценария
Post-Conditions	Учетная запись создана. Пользователь, для которого создавалась запись уведомлен.

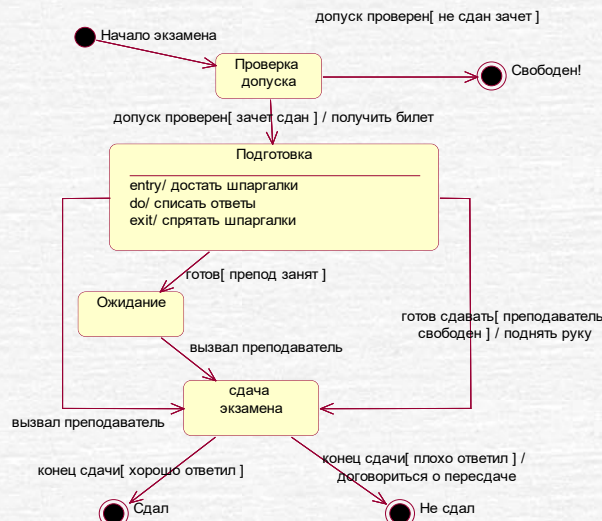
28

Діаграми станів

- Діаграми станів представляють скінчений автомат набором станів та переходів. Об'єкт можна описати з точки зору станів, у яких він може знаходитись, а також можливих переходів з одного стану в інший у випадку виникнення певних подій.
- State – стан об'єкту
- Event – событие, вызывающее переход
- Transition – перехід в новий стан
- Condition – умова переходу (true|false)
- Action – миттєва неперервна дія, яка супроводжує перехід
- Activity – діяльність, пов'язана зі станом

29

Приклад: здача екзамену

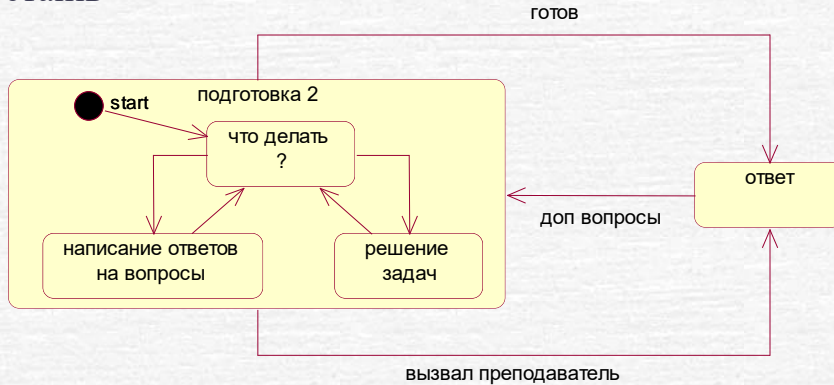


30

Приклад: вкладені стани

Застосування: групування станів і спрощення діаграми

Мають не більше одного початкового і кінцевого станів



31

Діаграми діяльності

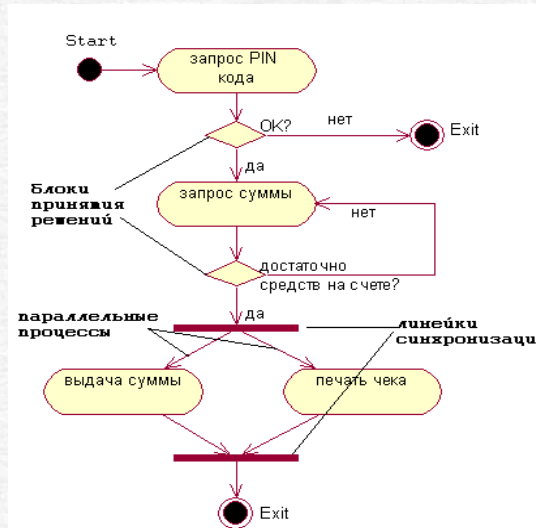
Діаграми діяльності надають можливість моделювання послідовності виконання дій у певному процесі. Діаграми діяльності аналогічні блок-схемам та можуть бути застосовані для представлення алгоритмів.

Зазвичай діаграми діяльності включають такі елементи:

- ☛ початковий стан
- ☛ діяльність
- ☛ перехід
- ☛ розгалуження
- ☛ кінцевий стан

32

Приклад: банкомат



33

Діаграма класів

Клас представляє множину об'єктів, яку об'єднує спільна структура та поведінка. Клас є абстракцією сутностей реального світу. Реальні екземпляри сутностей (instances) це так звані *об'єкти*.

Діаграма класів відображає статичну структуру системи у термінах класів та відношень між цими класами.

Графічним зображенням класу є прямокутник, розділений на три частини. Перша частина - це ім'я класу. Друга частина - список так званих атрибутів, третя частина - список операцій.

34

Приклад

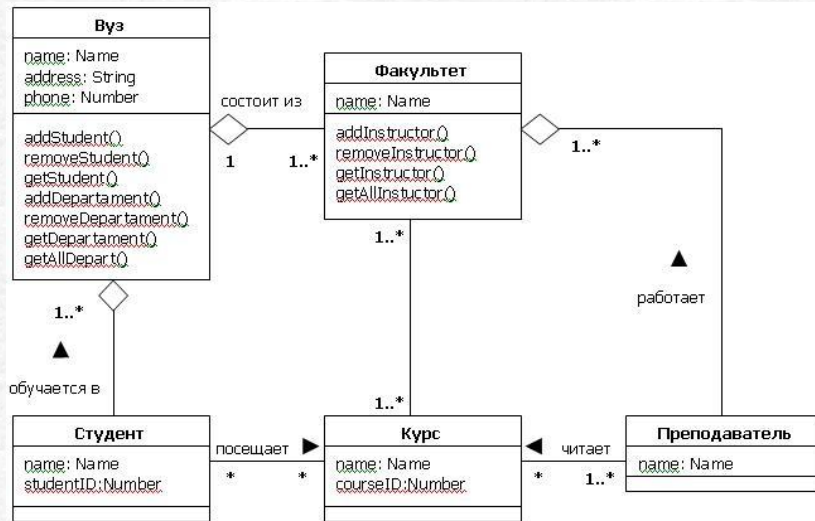


РИС 3.9

35

Діаграма об'єктів

Діаграма об'єктів представляє екземпляри класів (об'єкти). Графічне зображення об'єкту схоже на зображення класу. Ім'я екземпляру слід підкреслювати:



36

Діаграма послідовності

Діаграма послідовності - це графічне зображення послідовної взаємодії об'єктів. Діаграма послідовності має два виміри: по горизонталі розміщуються об'єкти, які взаємодіють між собою; вертикальний вимір - це вимір часу.

Типовими елементами діаграми послідовності є:

- об'єкт
- лінія життя
- повідомлення
- фокус управління
- повідомлення до себе

37

Диаграммы UML

Лінія життя - це пунктирна лінія, яка починається з піктограми об'єкту та спрямована униз.

Повідомлення, яке один об'єкт надсилає до іншого (приймача), обумовлює дію, яку потім виконує об'єкт-приймач. Графічно повідомлення зображується у вигляді стрілки, яка поєднує лінії життя двох об'єктів. Ім'я повідомлення - це фактично ім'я операції об'єкта-приймача.

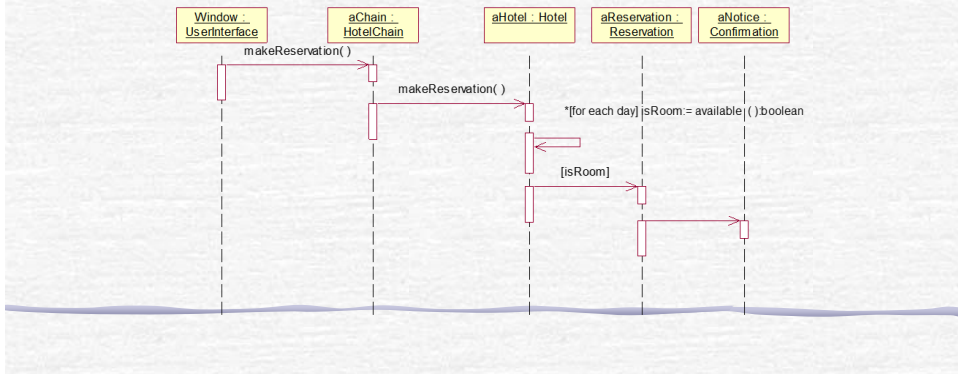
Фокус управління (focus of control) відображає період часу, протягом якого об'єкт виконує операцію або чекає на результат операцій інших об'єктів. Фокус управління зображають у вигляді прямокутника, який розташовано на лінії життя.

Повідомлення до себе є по суті викликом з однієї операції певного об'єкту іншої операції цього ж об'єкту.

38

Приклад

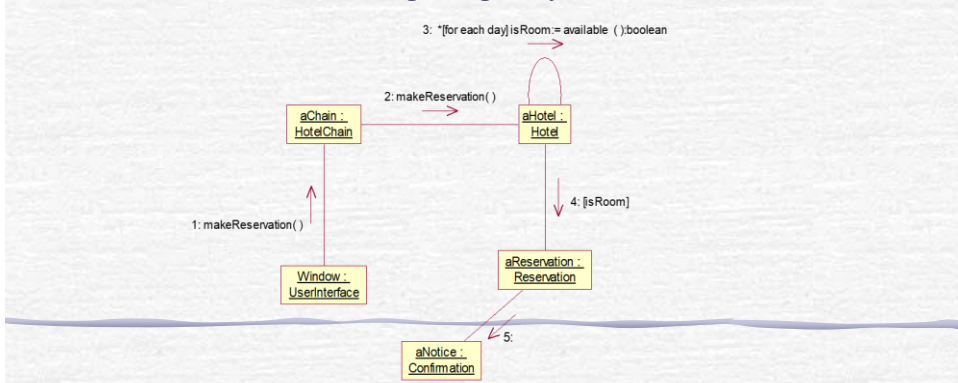
За допомогою діаграми послідовності можна представити взаємодію об'єктів моделі, упорядкованих за часом їхнього прояву. Діаграма послідовності має два виміри. Один – повідомлення, сигнал до виконання певної операції, яка повинна бути здійснена об'єктом, що прийняв повідомлення. Другий – лінія життя об'єкта (object lifeline), вертикальна лінія, що представляє існування об'єкта протягом певного періоду часу.



39

Діаграмами кооперації

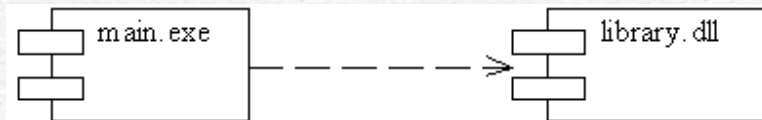
Діаграми послідовності семантично зв'язані з діаграмами комунікацій (у UML 1 - діаграмами кооперації). Фактично це альтернативне представлення діаграм послідовності. Діаграми комунікацій застосовують у тих випадках, коли необхідно відобразити велику кількість об'єктів, які взаємодіють між собою, а фактор часу має менше значення.



40

Діаграма компонентів

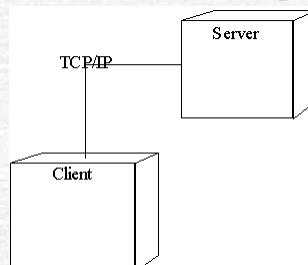
Діаграма компонентів описує фізичні компоненти програмного забезпечення (файли, динамічні бібліотеки та ін.) та відношення між ними у середовищі виконання програми. Не слід плутати компоненти з вихідними текстами та бібліотеками часу компіляції.



41

Діаграма розгортання

Діаграма розгортання показує фізичне розміщення компонентів апаратного забезпечення, необхідного для нормального функціонування комп'ютеризованої системи. Для окремих компонентів - вузлів (nodes) - можна вказувати, які саме програмні елементи на них розташовані.



42

Представлення

Всі аспекти, що моделюється не вдається описати з єдиної точки зору.

Моделювати складну систему слід з кількох різних точок зору, кожен раз беручи до уваги один аспект моделюється і абстрагуючись від інших.

Ця теза є одним з основоположних принципів UML.

43

Подання

Виділимо три подання:

- **подання використання** (що робить система корисного?);
- **подання структури** (з чого складається система?);
- **подання поведінки** (як працює система?).

44

Подання

Подання використання відповідає на питання, *що робить система корисного*.

Визначальною ознакою для віднесення елементів моделі до подання використання є явне зосередження уваги на факті наявності у системи зовнішніх кордонів, тобто виділення зовнішніх дійових осіб, взаємодіючих з системою, і внутрішніх варіантів використання, що описують різні сценарії взаємодії.

Описується діаграмою використання.

45

Подання

Подання структури відповідає на питання: *з чого складається система*.

Визначальною ознакою для віднесення елементів моделі до подання структури є явне виділення структурних елементів - складових частин системи - і опису взаємозв'язків між ними. Принциповим є чисто статичний характер опису, тобто відсутність поняття часу в будь-якій формі, зокрема, у формі послідовності подій і / або дій.

Описується діаграмами класів, а також, якщо потрібно, діаграмами компонентів і розміщення і, в окремих випадках, діаграмами об'єктів.

46

Подання

Подання поведінки відповідає на питання: **як працює система.**

Визначальною ознакою для віднесення елементів моделі до подання поведінки є явне використання поняття часу, зокрема, у формі опису послідовності подій / дій, тобто у формі алгоритму.

Описується діаграмами станів і діяльності, а також діаграмами взаємодії в формі діаграм кооперації та / або послідовності.

Література

1. И. Соммервиль. Инженерия программного обеспечения, 6 изд. – И.д. "Вильямс", 2002.
2. Г. Буч. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. Второе издание. – Бином, 1998.
3. Г. Буч, Дж. Рамбо, А. Джекобсон. UML. Руководство пользователя. – ДМК-Пресс, Питер, 2004.
4. G. Booch, J. Rumbaugh, I. Jacobson. The Unified Modeling Language Reference Manual – Second Edition, Addison-Wesley, 2004.
5. www.uml.org
6. www.wikipedia.org