

СУЧАСНІ ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ

**122 «Комп'ютерні науки»
КН-18**

2019 / 2020 навчальний рік

UML # 1.

Уніфікована мова моделювання

1. UML модель, UML мова.
2. Сутності UML .
3. Відношення UML.
4. Діаграми UML.

https://github.com/eabshkvprof/2020_Mod_Prog_Tech

Уніфікована мова моделювання
UML (*Unified Modeling Language*) -
графічна мова об'єктного моделювання
при розробці програмного забезпечення,
для моделювання бізнес-процесів,
системного проектування та
відображення організаційних структур.



Object Management Group (OMG) <https://www.omg.org/>

Перша версія UML 1.0 → 1997 (Буч Г., Рамбо Д., Якобсон А.)

UML 1.4.2 → стандарт ISO/IEC 19501:2005

UML 2.4.1 → стандарт ISO/IEC 9505-1, 19505-2.

UML 2.5 → 2015

Уніфікована мова моделювання

An OMG® Unified Modeling Language® Publication



OMG® Unified Modeling Language® (OMG UML®)

Version 2.5.1

OMG Document Number: formal/2017-12-05

Date: December 2017

Остання версія 2.5.1 <https://www.omg.org/spec/UML/About-UML/>



Уніфікована мова моделювання

UML надає стандартний спосіб написання проектної документації на системи, включаючи **концептуальні аспекти** (бізнес процеси, функції системи) + **конкретні аспекти** (вираження мов програмування, схеми баз даних і повторно використовувані компоненти)

UML - не є формальною мовою.

UML - не є мовою програмування.

UML - не є методологією.

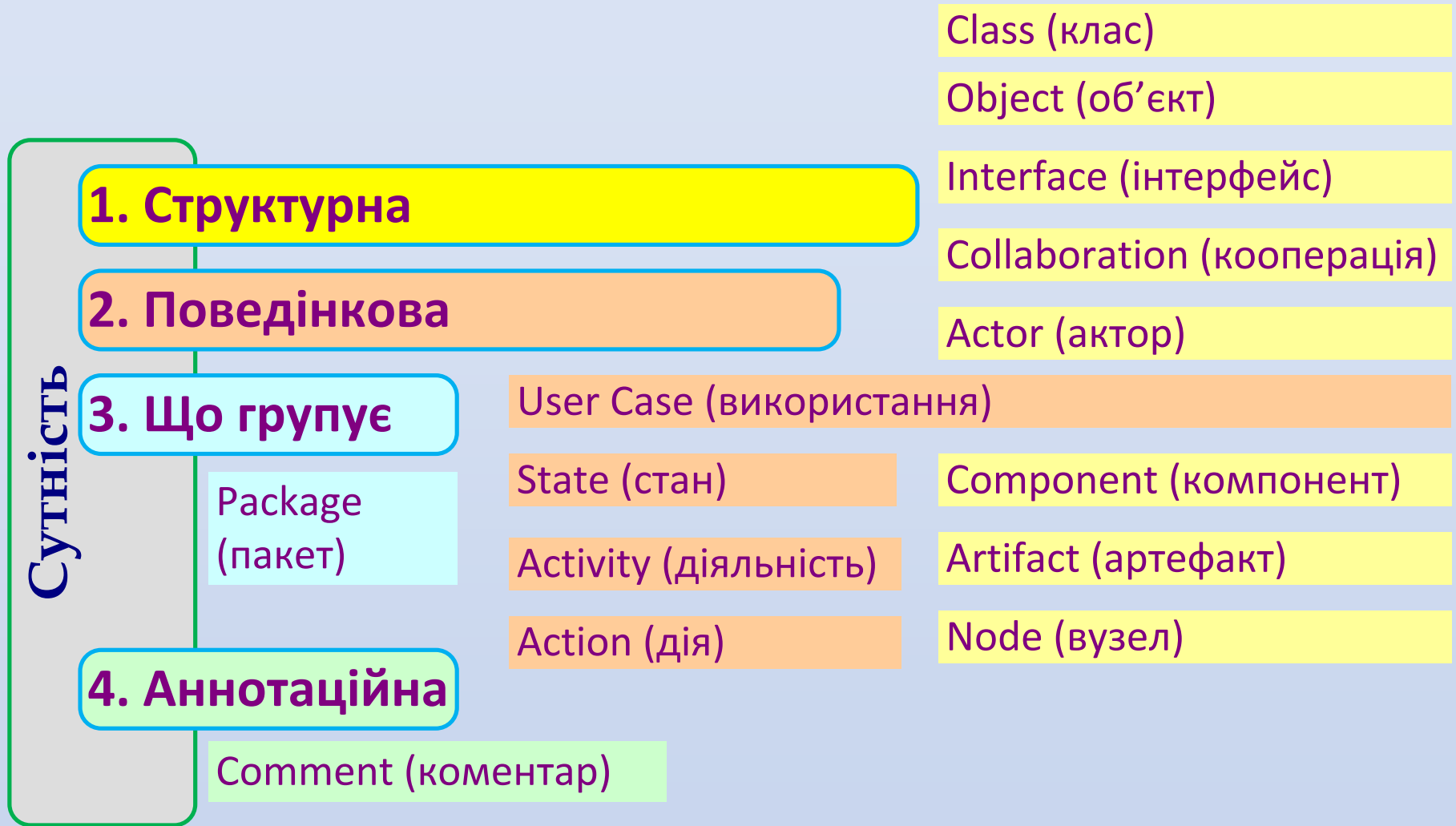
UML - не є процесом.

UML.

UML модель – це граф (навантажений гіпер - граф), в якому вершини і ребра навантажені додатково інформацією і можуть мати складну внутрішню структуру. Вершини цього графа - **сутності**, ребра - **відношення**.



UML. Сутності



UML. Структурні сутності

Class / Клас - множина об'єктів із загальними атрибутами, що визначають стан і операціями, що визначають поведінку

new_class

Object / Об'єкт - сутність, що володіє унікальністю і інкапсулює в собі стан і поведінку (екземпляр класу)

Object 25 : New_Class

UML. Структурні сутності

Interface / Інтерфейс -

іменована множина операцій, що визначає набір послуг, які можуть бути запитані споживачем і надані постачальником послуг.

«interface»
new_interface

Component / Компонент -

модульна частина системи з чітко визначеним набором необхідних інтерфейсів та інтерфейсів, що надаються.

Data Base 1

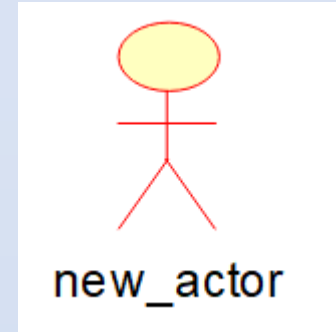
Cooperation / Кооперація -

сукупність об'єктів, які взаємодіють для досягнення певної мети.

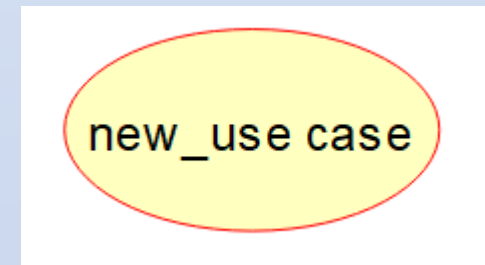
Visitor

UML. Структурні сутності

Actor / Актор - сутність, яка перебуває поза модельованої системи і безпосередньо взаємодіє з нею (дійова особа, користувач).



Use case / Використання - безліч сценаріїв, об'єднаних за деякими критерієм, які описують послідовність дій, що виконуються системою, та доставляють значимий для деякого Актору результат



(варіант використання, прецедент, сценарій поведінки).

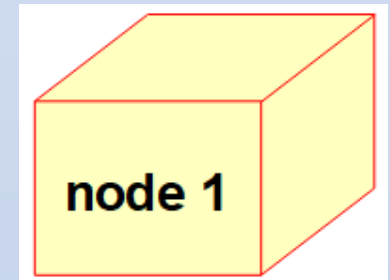
! також Поведінкова сутність

UML. Структурні сутності

Artefact / Артефакт - елемент інформації який використовується або породжується в процесі розробки програмного забезпечення. Фізична одиниця реалізації, що отримується з елемента моделі, наприклад з класу або компоненту.

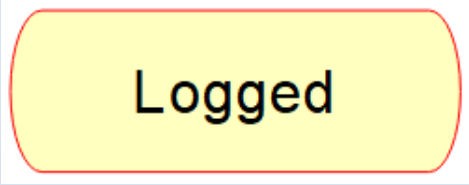


Node / Вузол - обчислювальний ресурс , на якому розміщуються та , при необхідності, виконуються артефакти



UML. Поведінкові сутності

State /Стан – деякий період життєвого циклу об'єкту, перебуваючи в якому об'єкт, задовольняє деякій умові та здійснює власну діяльність, або очікує настання деякої події.



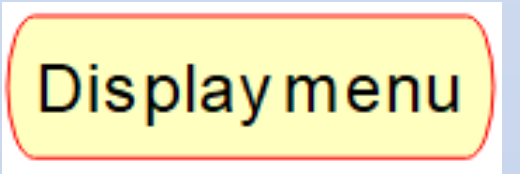
Logged

Activity / Діяльність – частка стану, яка характеризується тривалими за часом не атомарними обчисленнями.



new activity

Action / Дія – примітивне (атомарне) обчислення



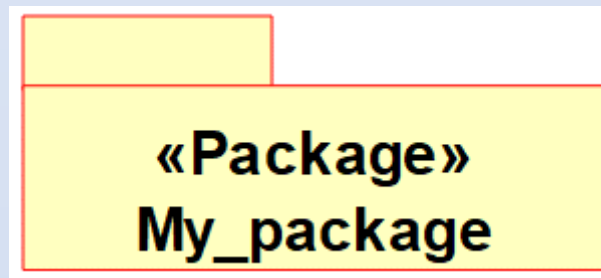
Display menu



A := in (text)

UML. Групова сутність

Package / Пакет - група елементів моделі, у тому числі і пакетів.



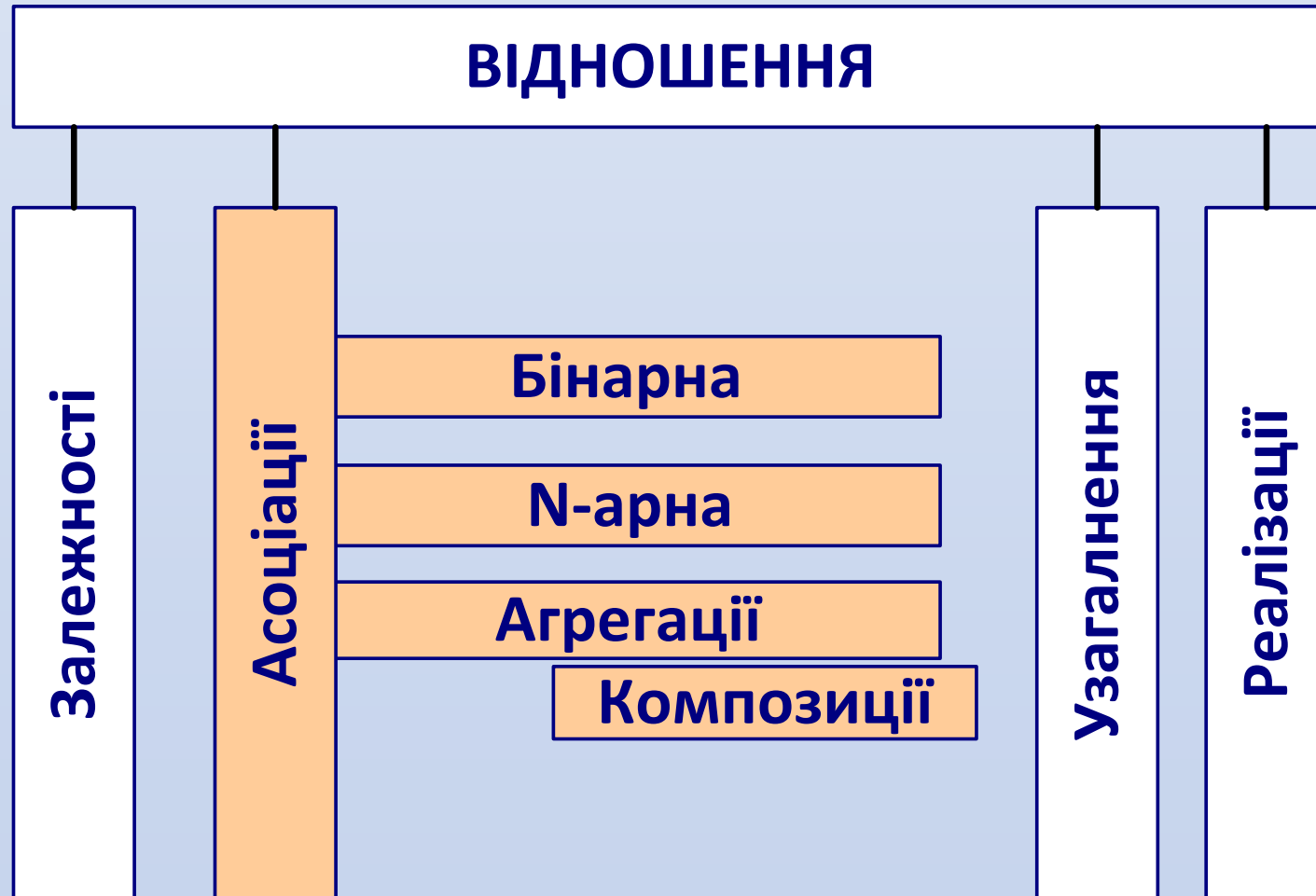
UML. Анотаційна сутність

Comment / Коментар - деякий текстовий опис



UML. Відношення

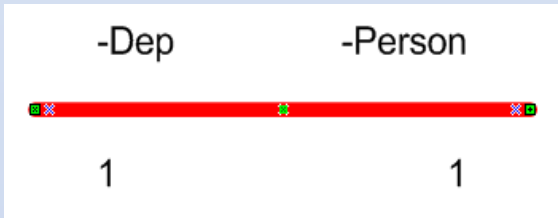
Відношення – відображення семантичного зв'язку між сутностями



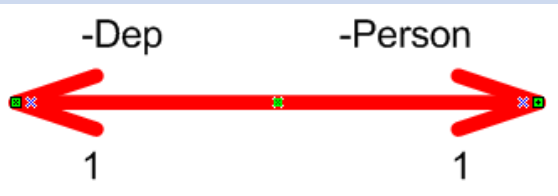
UML. Відношення асоціації

Асоціації – показує що одна сутність безпосередньо пов'язана з іншою (або іншими).

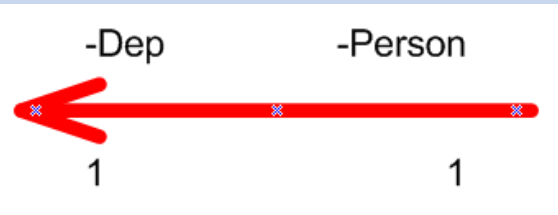
Дві сутності
- Бінарна
Без напрямку



Двонаправлена



Однонаправлена



Множинність асоціації

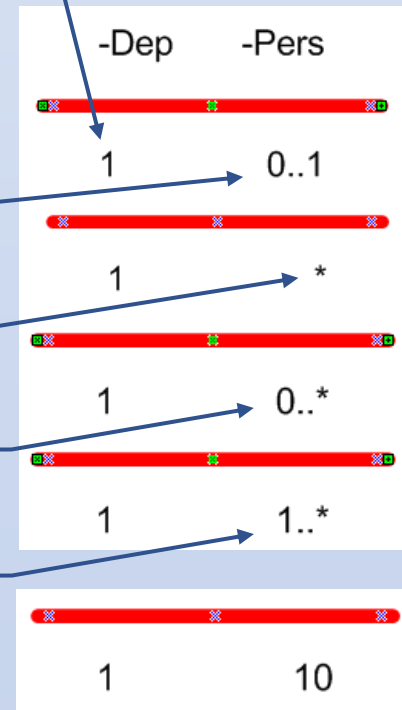
Одне входження

Одне, або відсутнє

Багато

0, або багато

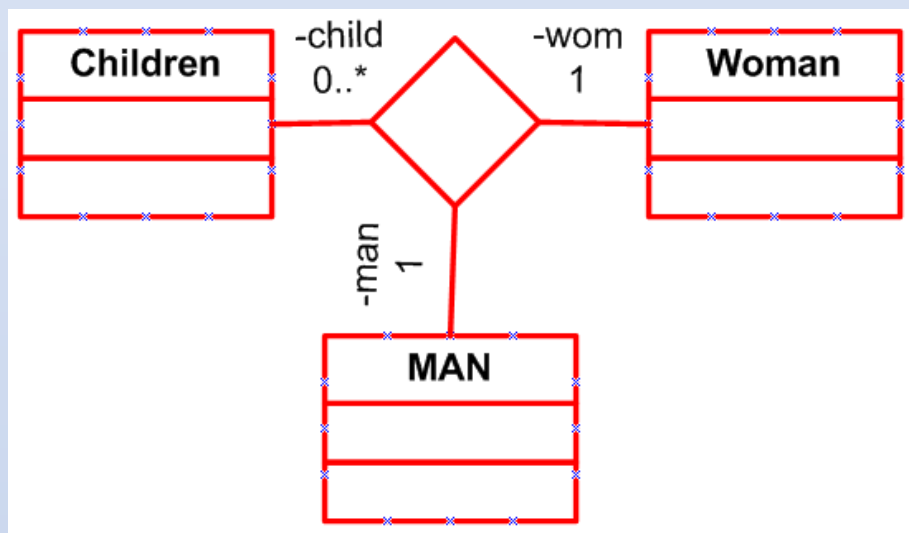
1, або багато



UML. Відношення асоціації

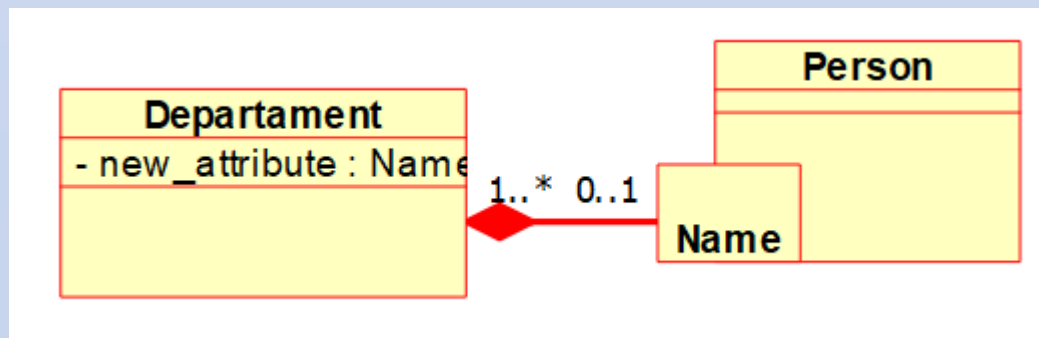
Асоціації – показує що одна сутність безпосередньо пов'язана з іншою (або іншими).

N-арна асоціація – пов'язує декілька сутностей



Агрегація

Композиція



UML. Відношення асоціації

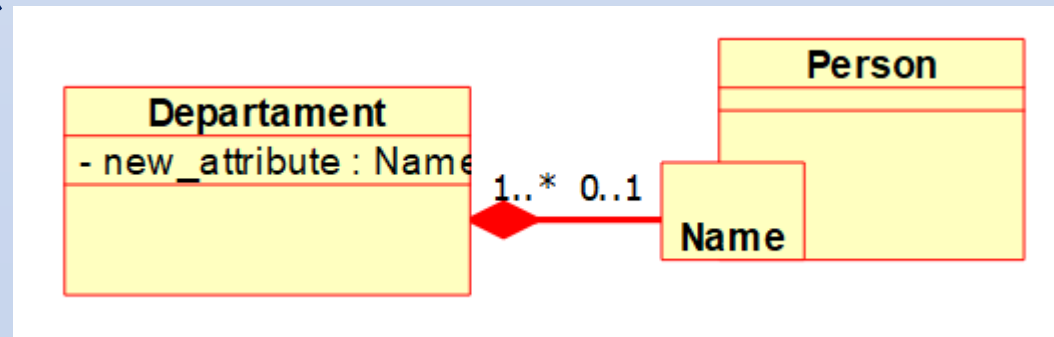
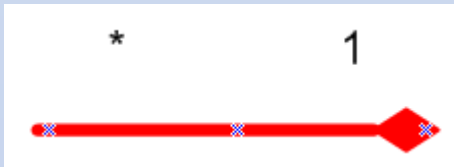
Асоціації – показує що одна сутність безпосередньо пов'язана з іншою (або іншими).

Не мають строгого визначення :

Aggregation / Агрегація – визначає зв'язок «ціле-частка (is part of)» між сутностями.



Composition / Композиція –
"сильна" агрегація (is
entirely made of)

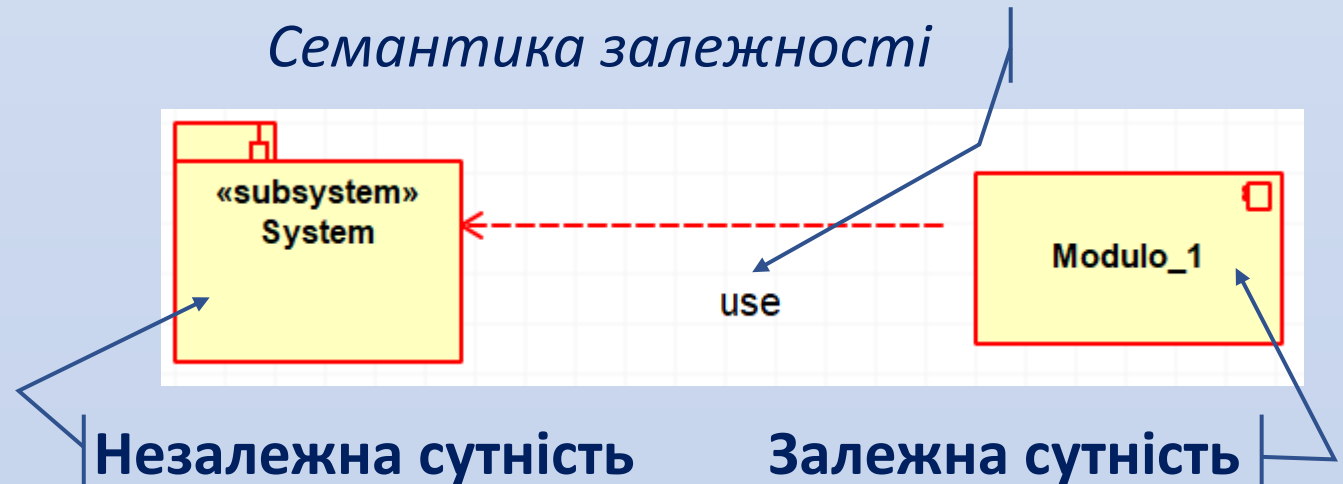


UML. Відношення залежності

Dependency / Залежність – загальний тип відносин між двома сутностями. Показує що зміна незалежної сутності впливає на залежну сутність (uses temporarily).

Незалежна сутність може і не знати, що її використовує інша сутність. Залежна сутність не може не знати, від чого вона залежить.

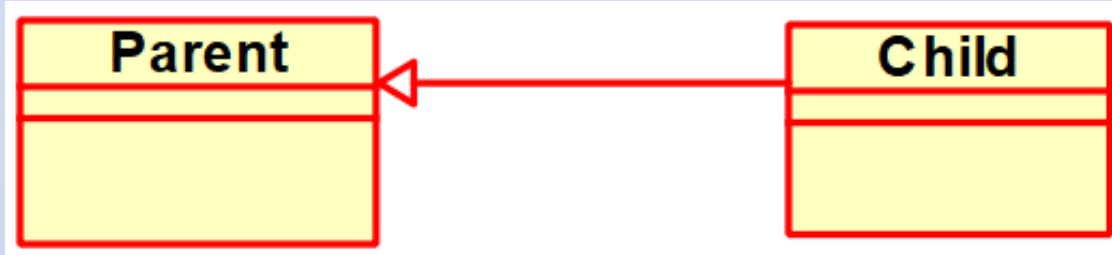
Напрямок стрілки від залежної сутності до незалежної.



UML. Відношення узагальнення

Generalization / узагальнення -

відношення типу «загальне-спеціальне», коли об'єкт спеціалізованого елемента (нащадок, child) може бути представлений замість об'єкта узагальненого елемента (батька, предку, parent), але не навпаки.



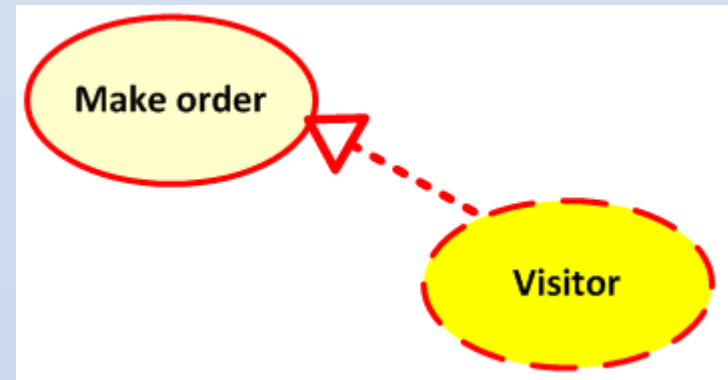
Child успадковує
структуру і
поведінку свого
parent.

UML. Відношення реалізації

Realization / реалізації -

відношення між класифікаторами, з яких один визначає зобов'язання, а інший гарантує їх виконання. Наприклад:

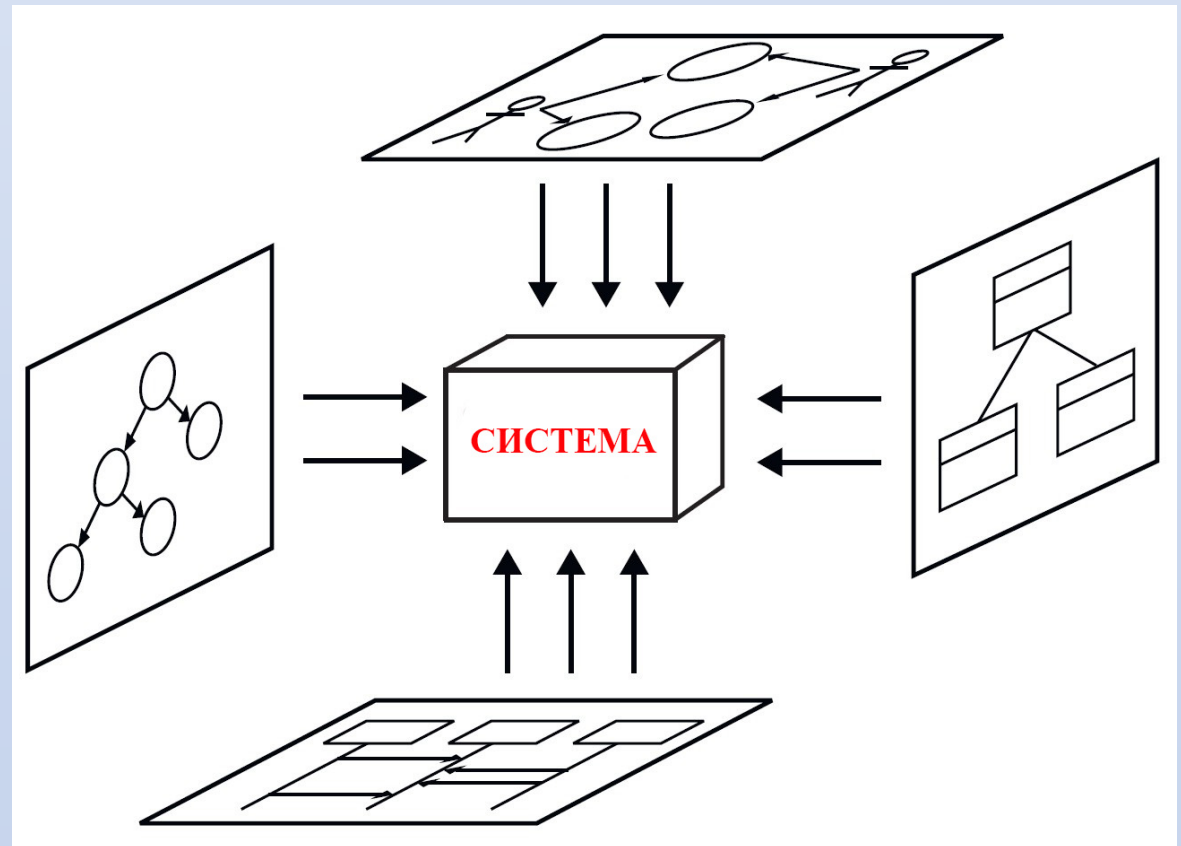
- між інтерфейсами та класами (компонентами) що їх реалізують,
- між прецедентами та коопераціями, що їх реалізують



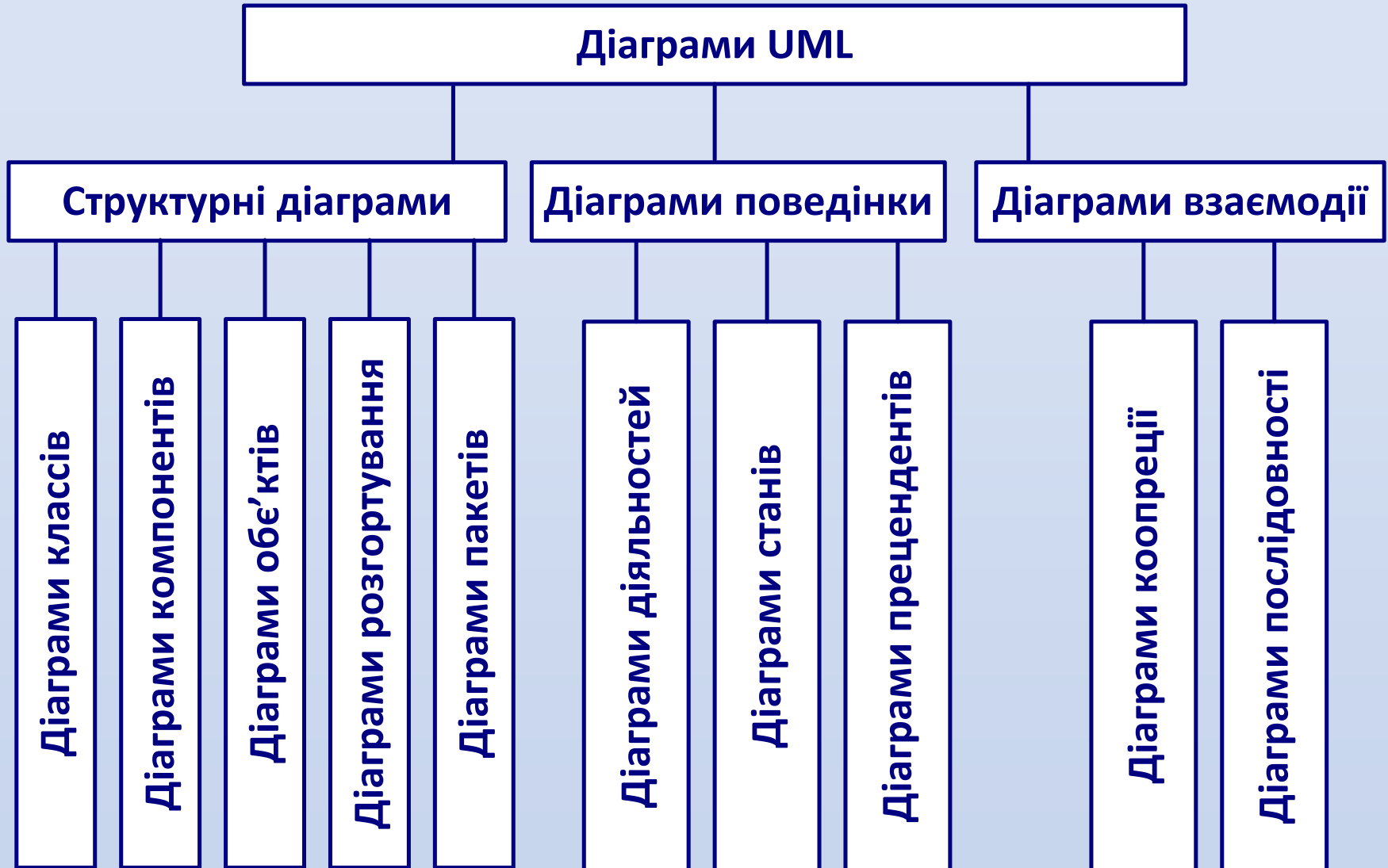
UML. Діаграми

Діаграма (diagram) - графічне представлення сукупності елементів моделі у формі зв'язного графа, вершинам і ребрам (дугам) якого приписується певна семантика

Діаграма –
“погляд” на
систему, що
моделюється



UML. Діаграми

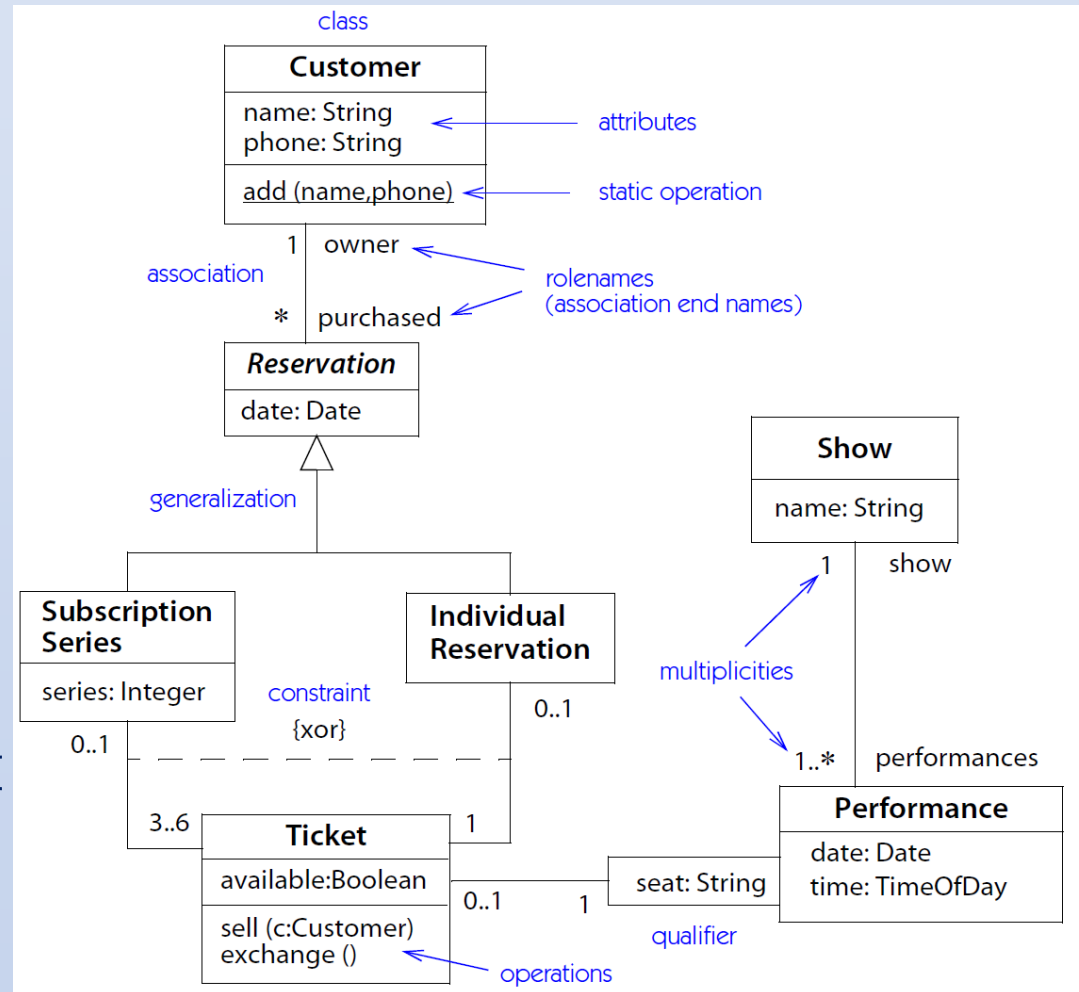


UML. Діаграма класів

Class diagram – базовий засіб опису структури системи. Зображає класи, інтерфейси об'єкти та кооперації та відношення між ними

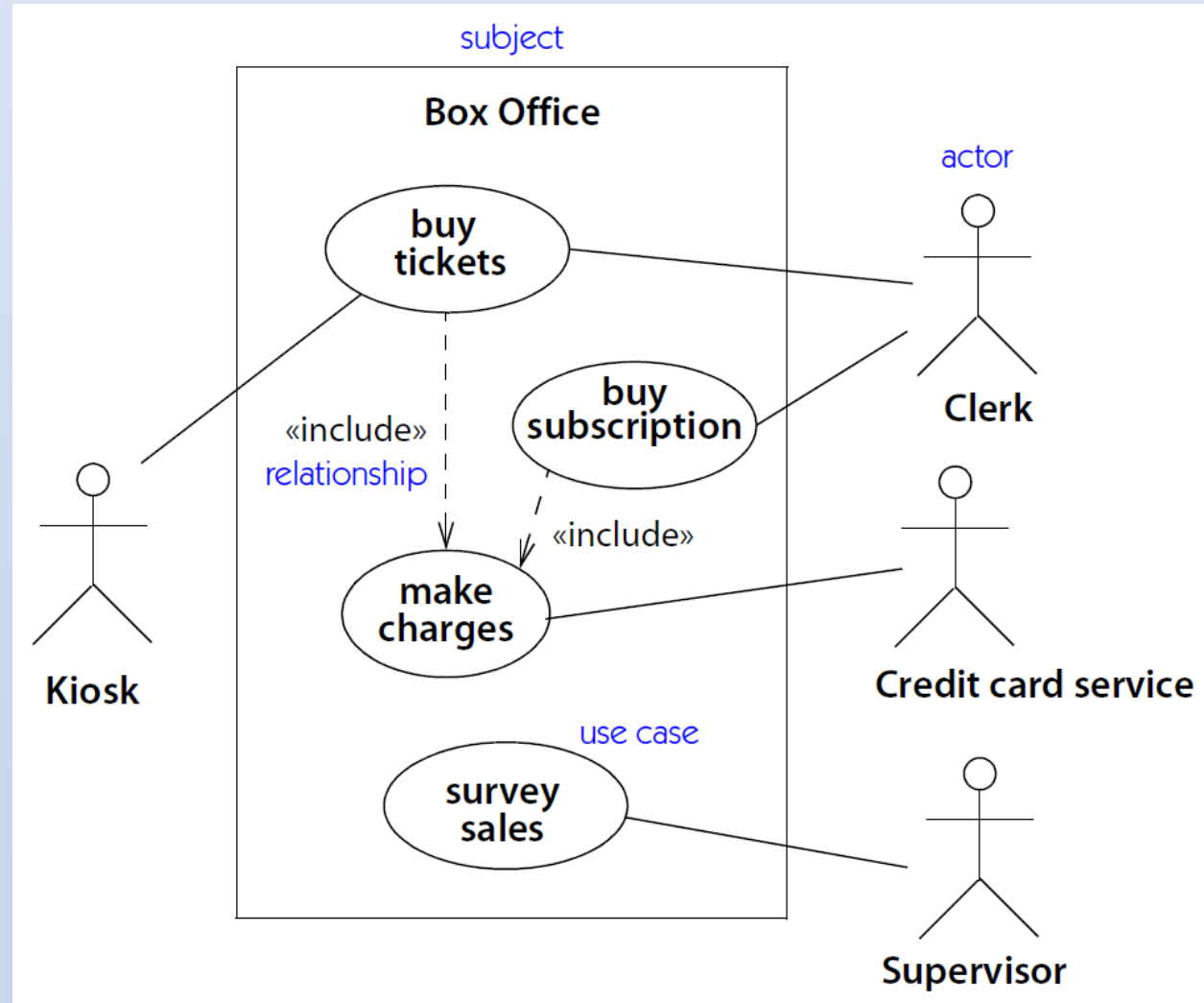
Вказуються відношення, в першу чергу:

- асоціації між класами,
- узагальнення між класами,
- залежності між класами та класами та інтерфейсами.



UML. Діаграма прецедентів

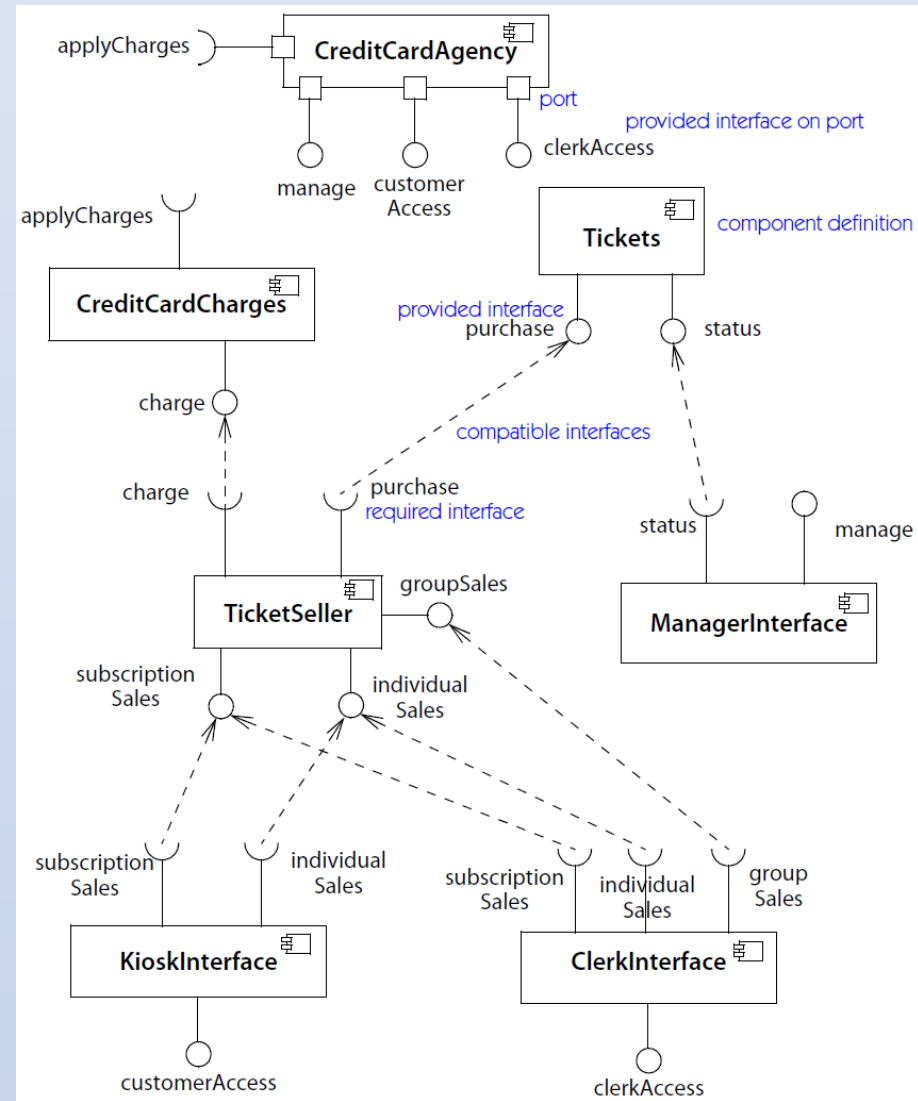
User case diagram
зображають акторів та
прецеденти та
відношення
між ними



UML. Діаграма компонентів

Component diagram

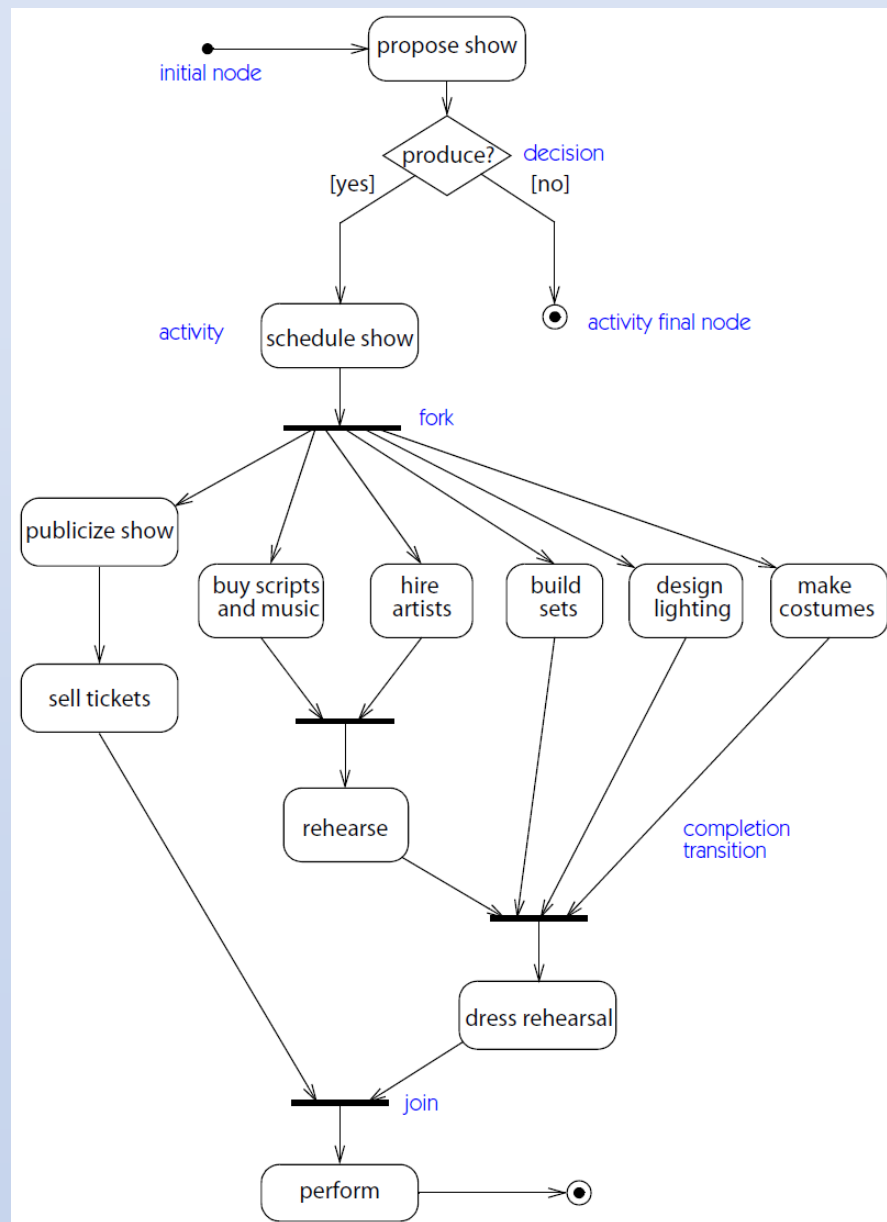
- зображає сукупність компонентів системи та існуючі між ними залежності.



UML. Діаграма діяльності

Activity diagram

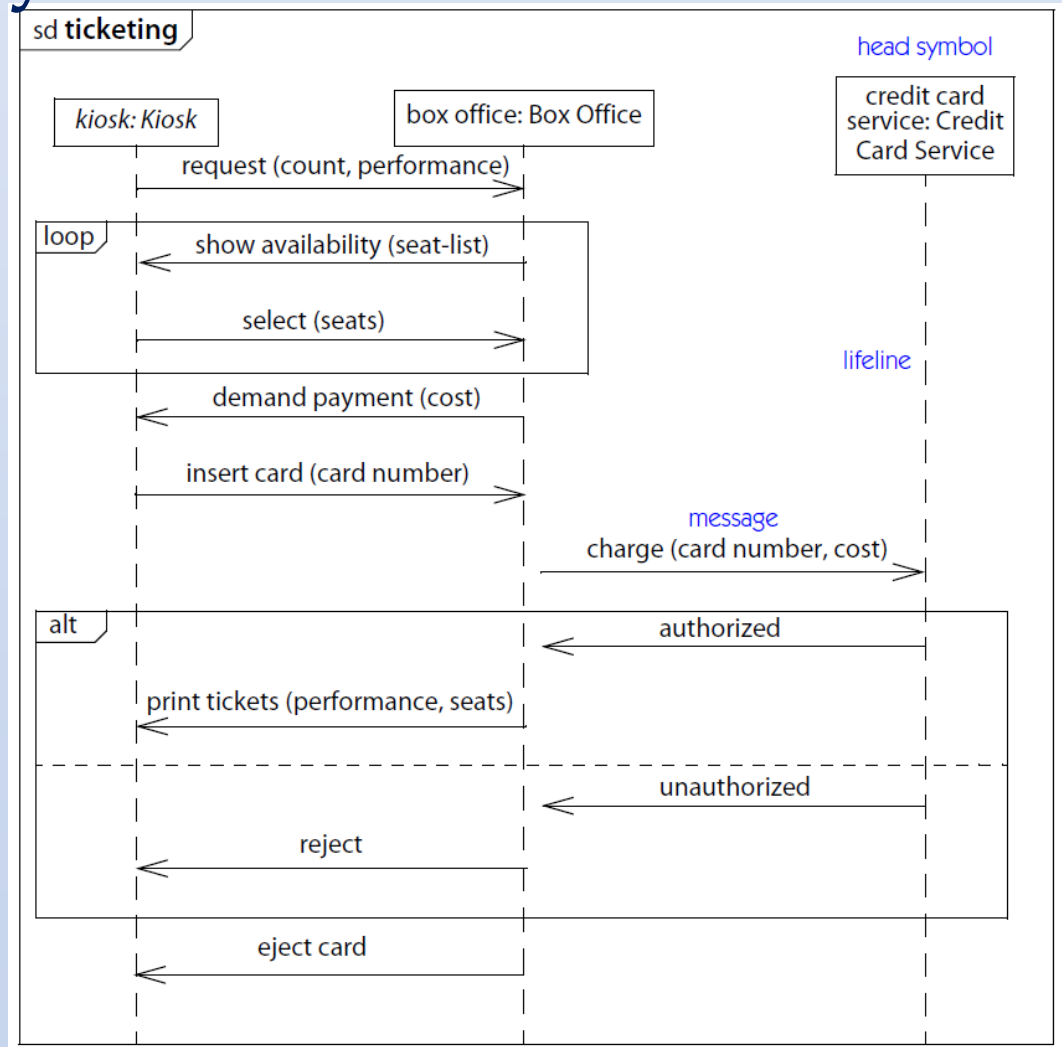
- зображає перехід потоку керування від одного виду діяльності до іншої в системі.



UML. Діаграма послідовності

Sequence diagram

- Відображає часову
упорядкованість
повідомлень



UML. Діаграми. Вимоги

- Кожна діаграма повинна служити закінченим представленням відповідного фрагмента модельованої предметної області.
- Всі сутності на діаграмі моделі повинні бути одного концептуального рівня.
- Вся інформація про сутності повинна бути явно представлена на діаграмах.
- Діаграми не повинні містити суперечливої інформації.
- Діаграми не слід перевантажувати текстовою інформацією.
- Кожна діаграма повинна бути саме достатньою для правильної інтерпретації всіх її елементів і розуміння семантики всіх використовуваних графічних символів.

Додаткова ЛІТЕРАТУРА

- **Томашевський О.М., Цегелік Г.Г., Вітер М.Б., Дудук В.Ш. Інформаційні технології та моделювання бізнес-процесів. Навч. посіб. – К.: «Видавництво «Центр учбової літератури», 2012. – 296 с.**
- **Карпенко М.Ю., Манакова Н.О., Гавриленко І.О. Технології створення програмних продуктів та інформаційних систем. Навч. посіб. - Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім.О.М. Бекетова, 2017. – 93 с.**
- **Алексенко О.В. Технології програмування та створення програмних продуктів. Конспект лекцій. – Суми, Сумський державний університет, 2013. – 133с.**

Рекомендована ЛІТЕРАТУРА

- **Дудзяний І.М. Об'єктно-орієнтоване моделювання програмних систем: Навчальний посібник.** – Львів, Видавничий центр ЛНУ ім. Івана Франка, 2007. – 108с.
- **Rumbaugh J., Jacobson I., Booch G. The Unified Modeling Language. Reference Manual.** – NY.: Addison-Wesley, 2005 .- 721 p.

Посилання
Object Management Group (OMG)
<https://www.omg.org/>

Контрольні запитання

- Визначте поняття UML мови та UML моделі. Наведіть переваги використання UML.
- Надайте перелік та визначте призначення сутностей, які використовує мова UML. Наведіть відповідні графічне позначення сутностей.
- Надайте перелік та визначте відношення між сутностями, які використовує мова UML. Наведіть відповідні графічне позначення відношень.
- Визначте поняття діаграми UML, надайте перелік найбільш вживаних діаграм та їх призначення.

The END
Mod 3. Lec 1.