Type of UML

2-class diagram

✖ NaSeem Aljabri

1- class = Properties + method + instantation

1- **CLASS NAME**
2- Properties/attributes
3- Methods / operations

→ most have an identical meaning.

→ be varibles

2- syntax of attributs:- *type of accessibility → ① + (bablic)

→ ③ # (Protected)

→ ② - (Private)

3- syntax of operation :-

4- Relationship between classes :-

1- Association :-

يمثل خط مستقيم (كل الكلاسين توصل لبعض)

ex :-

2- Aggregation :-

كل كلاس مستقل عن الاخر

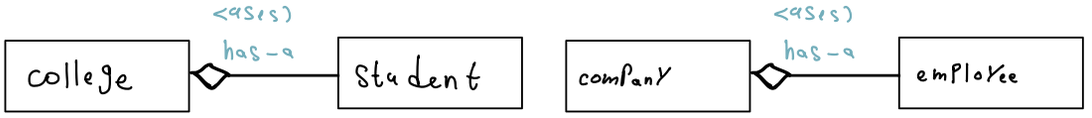
* weak relationship

ولو احذف واحد الاخر يبقى

ex :-

لو حذفنا college سيبقى Student

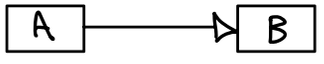
هنا لو راح الموظف تبقى الشركة



4- Inheritance :-

superclass > بيكون عندنا *
subclass

* وبنستخدم مع هاذا النوع العلاقات العددية (multiplicity)



* تمثيله :-

3- composition :-

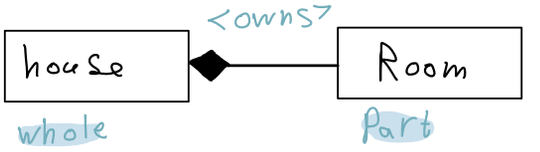
عكس Aggregation

* strong relationship

لو احذف كلاس الاخر بنحذف معاه

ex :-

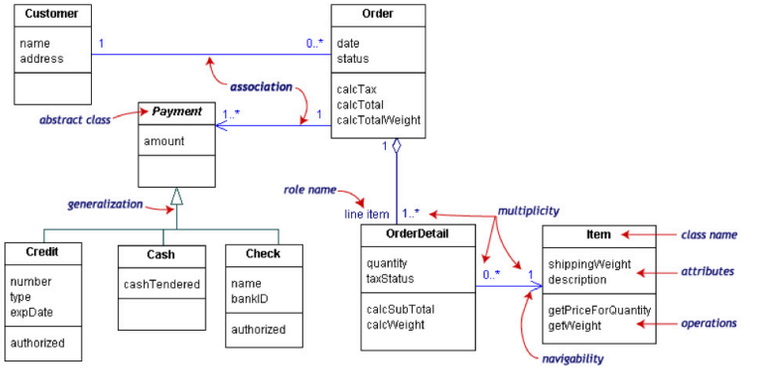
لو حذفنا كلاس house ما حيكون Room عندنا



* multiplicity :-

1	Exactly one
0 .. 1	zero or one
N (example : 8)	N (integer)
M .. N (example : 3 .. 7)	From M to N (integer)
***** **0 .. ***	From 0 to many
1 .. *	From 1 to many

* examples :-



Class identification

most specify a type the engineering (used for designing and analyzing software system)
Tow types:

- Forward engineering :
  - -Can find the classes for new software system starting with requirements
  - -first use in modeling and than implementation
- Reverse engineering:
  - -identify the classes in existing system
  - -involves analyzing code,documentation
  - -Difficult knowing the relationship between classes

Issues with class identification

- Definition of the system boundary:
- Classes/Objects are not just found by taking a picture of a scene or domain:

Finding Participating Objects in Use Cases (Abbott’ s Technique)

- Nouns : object and classes
- Verbs : operations

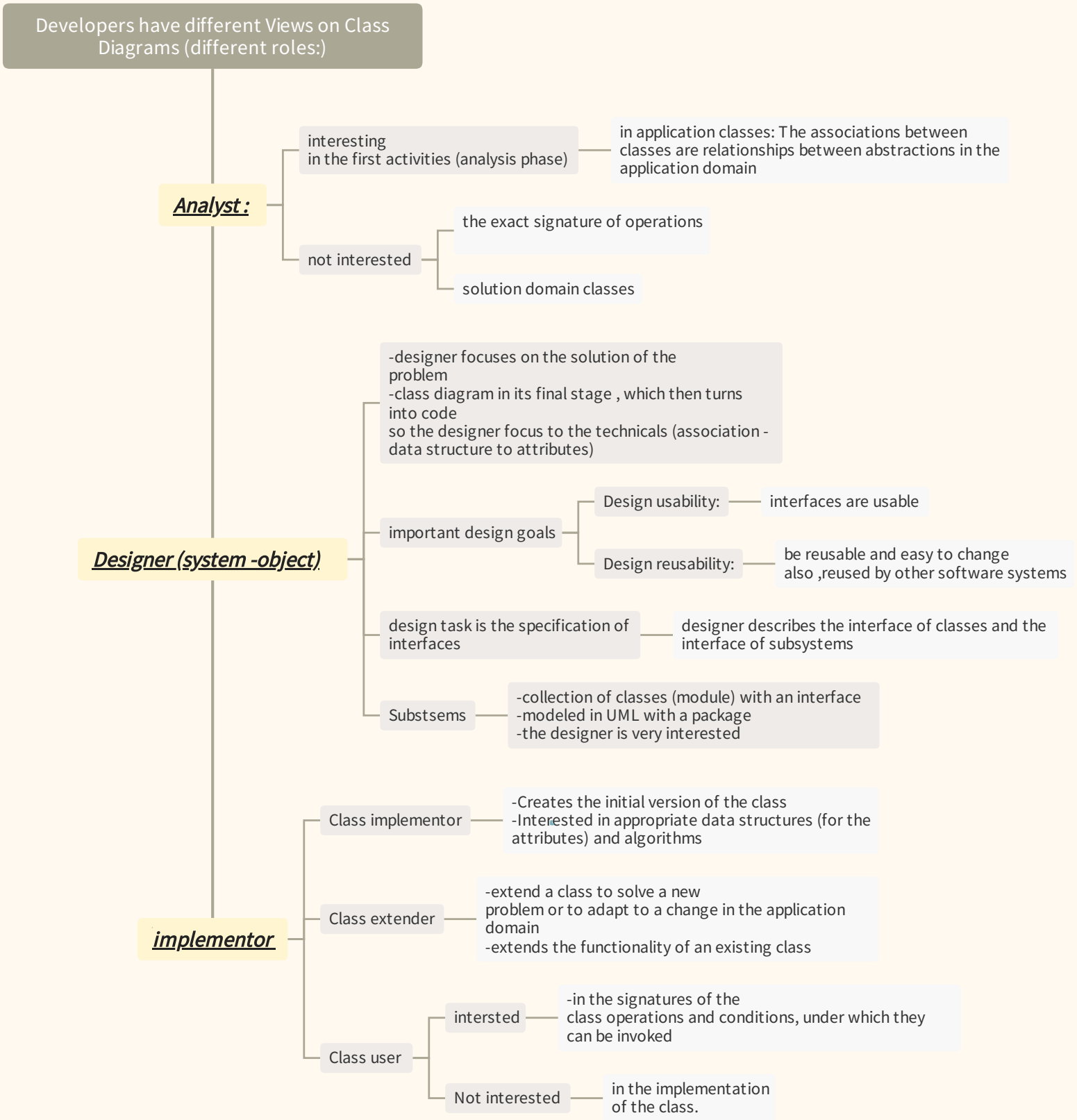
Class diagram describes the static properties of the system

Uses class diagram:

- *application domain*
  - During requirements analysis
- *developers*
  - During
    -system design
    -object design
    -implementation

Doesn’ t use class diagram

- *client*
  - focus more on management
- *end user*
  - interested in functional


Developers have different Views on Class Diagrams (different roles:)
Analyst :
interesting
in the first activities (analysis phase)
in application classes: The associations between classes are relationships between abstractions in the application domain
not interested
the exact signature of operations
solution domain classes
Designer (system -object)
-designer focuses on the solution of the problem
-class diagram in its final stage , which then turns into code
so the designer focus to the technicals (association - data structure to attributes)
important design goals
Design usability:
interfaces are usable
Design reusability:
be reusable and easy to change
also ,reused by other software systems
design task is the specification of interfaces
designer describes the interface of classes and the interface of subsystems
Substsems
-collection of classes (module) with an interface
-modeled in UML with a package
-the designer is very interested
implementor
Class implementor
-Creates the initial version of the class
-Interested in appropriate data structures (for the attributes) and algorithms
Class extender
-extend a class to solve a new problem or to adapt to a change in the application domain
-extends the functionality of an existing class
Class user
intersted
-in the signatures of the class operations and conditions, under which they can be invoked
Not interested
in the implementation of the class.

	Analysis model	object Design model
main stakholder	- end user - customer - analyst	- class specifiers - class extenders - class user - class implementor
The class diagram contains	only Application domain	Application domain as will Solution domain
the basis for comunication	btween analyst, Application domain and end user	between designers and implementors.