

System Design UML Diagrams

* What is UML Diagrams?

→ Suppose we have an idea of making an application & I want to tell it to my friend about how well make an application. So we have two options about how to tell ~~the~~ about this app to my friend :-

- (i) By writing a large paragraph about a working of the application & its design principles used. → inefficient / boring way to convey the idea.
- (ii) Best way is to convey anything is by representing it in a diagram.

∴ A UML (Unified modeling language) is a visual representation of a system's structure & behaviour. UML is widely used in S/w engg. to model, visualize & document the design of s/w systems. It provides a standardized way to visualize the architecture, design patterns, & relationships b/w different components of a system.

We'll learn UML diagram
Structural diagram i.e.
from class diagram \Rightarrow S.V.V.U.M.B.

Page No.:
Date: / /

Temporary types of UML Diagrams:

(i) Structural Diagrams: Also k/s static diagrams
Tells us about the structure of our application i.e. its type of components & how they are connected to each other

(ii) Behavioral Diagrams: Also k/s dynamic diagrams
Tells us about the behaviour of our application i.e. how the components interact with each other

We don't need to study all 14 UML diagrams instead we only need to study 4 UML diagrams i.e. 1 from each (structural & behavioral), the rest 12 UML diagrams are use-case specific, that's why we'll not study them.

Class Diagram: Tells us that what type of classes will be present in our application & how they are associated from each other.

Class Structure

How to represent a class in our UML

Diagram, class name

Eg = class Car

Car

↳ className

variables: string brand, model;
int engineCC;
protected void startEngine();

+brand: string
+model: string
+engineCC: int

↳ Characters/variables

methods or behavior:
void stopEngine();
public void accelerate();
void break();

+startEngine(): void
+stopEngine(): void
+accelerate(): void
+break(): void

↳ Behaviour/methods/func

This is over UML diagram but it must also represent access modifier.

How to represent access modifier logically in our UML diagram?		Access modifier			within class	child class	outside class	Representation
public		✓	✓	✓	✓	✓	✓	(+)
protected		✓	✓	✓	✓	✓	✗	(#)
private	more contradict	✓	✓	✗	✗	✗	✗	(-)

② How to define abstract & concrete class

⇒ (i) Abstract class = mtl. yisnein virtual meth. hai, i-e size methods jo declare class mein kya hai par define child class krega

→ to is declare kro. hai to wo class ke box ke upper << abstract >> lekhao

(ii) Concrete class = represent kro ke size class ke box ke upper kuch likhe ka surat nahi hse

General overview ~~abstract~~

Class Name
(f) var1 : datatype
(f) var2 : datatype
var3 : datatype
(f) method1 : datatype
method2 : datatype
(f) method3 : datatype

Association

Types of associations

↓
[Class association]

k/s inheritance
&
(also k/s is-a relation)

e.g. ManualCar is also k/s has-a relation
inherited from class
there is a relationship
b/w them

↓
[Object associat.]

[Simple
Association]

↓
[Aggregation composite]

Combining we call
it as composition only
as to represent them, we
have only one method
~~from~~ programmatically, but
it's different theoretically only

① Inheritance

Class A<

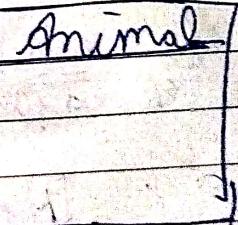
method1()

How to

represent it?

Using ^{Closed} arrow (→)

≡
}
}



Class B extends class A<

method2()

≡
}

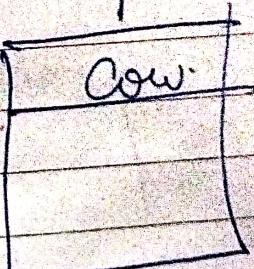
main()

B b = new B();

b.method1();

b.method2();

Cow

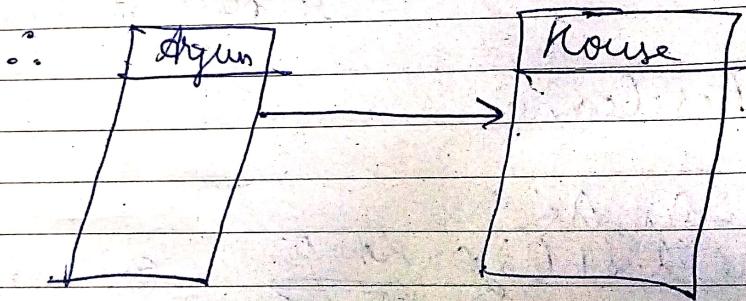


① Composition
↳ (has-a relation)

most used in OOD. More than inheritance
(i) Simple Association: Weakest form of inheritance
b/w two classes.

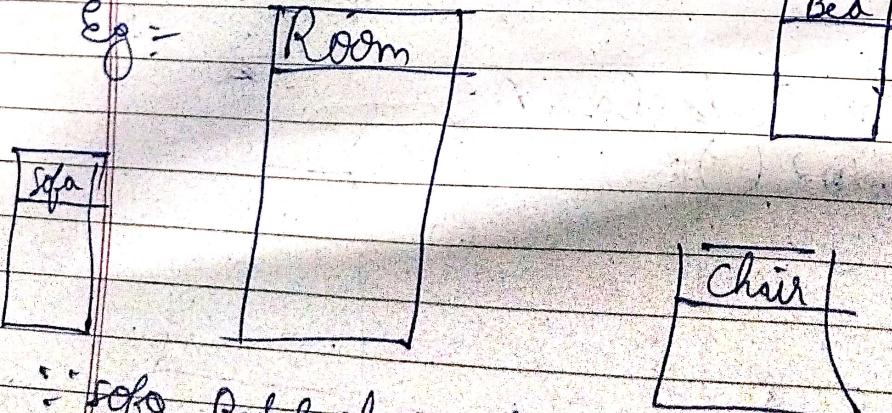
Eg: In real life we can say that
Arijun lives in a house
Arijun has a house
↳ No complex relationship

For representing simple association in UML
diagrams we use (→) open arrow.



(ii) Aggregation: Two objects are more interlinked
with each other rather than simple association

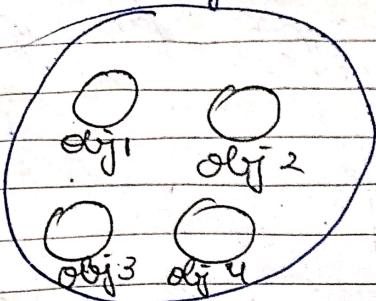
Eg:



Sofa, Bed & chair all are present in a room
we can say that Room has sofa, bed & chair

∴ In aggregation one main object will be present & will act as a container for other objects inside it.

Date: / /



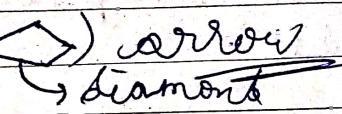
Object



⇒ Aggregator relationship

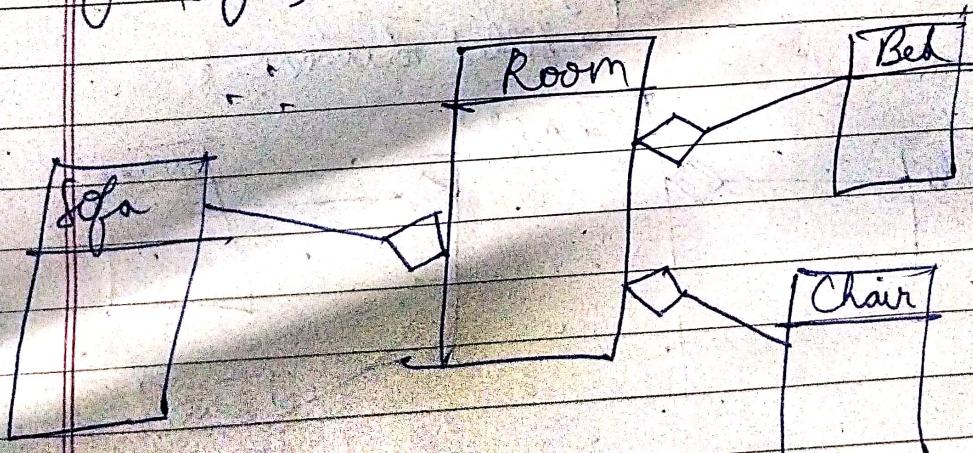
- Stronger than simple association but weaker than composition.

Representation for aggregator relationship in UML diagram

We use () arrow
diamond

We'll put the diamond part from small objects (obj1, obj2 etc) to big object/container

∴ Room will act as a container for sofa, chair & bed.



Main b^o sofa, bed & chair is a part of room for sure but they can also exist individually.

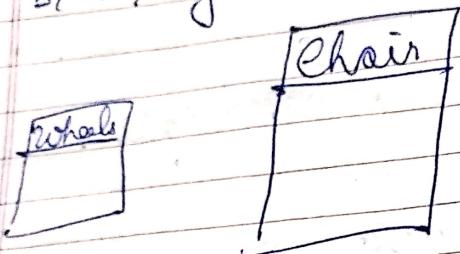
Date: 10/10/2023
Page No.: 10

Now, how

aggregation & co

(iii) composition = strongest form of relation b/w any two objects.

forms

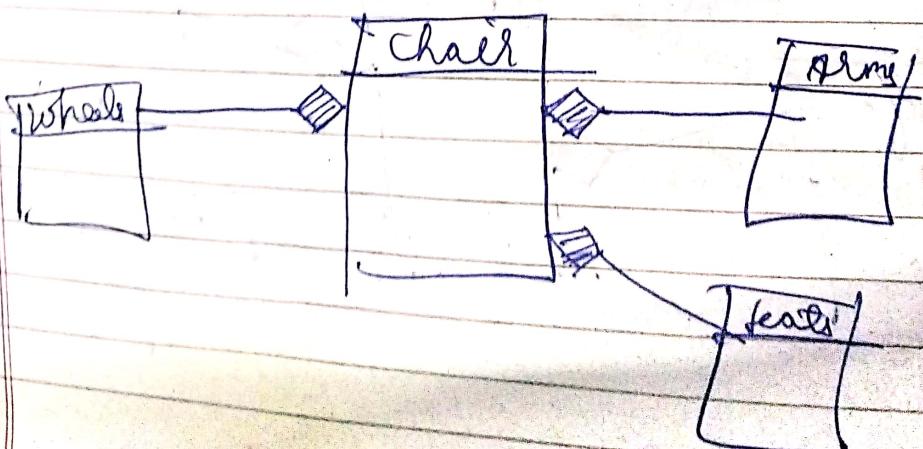


has
i.e. chair ~~contains~~ arms, seat & wheels
i.e. also arms, seats & wheels cannot exist independently or individually w.r.t chair.

∴ We can say that chair is made up of wheel, arms & seat.
→ This is only a/s composition relations.

Representation of composition relationship :-
We use (—)

→ filled diamond.



Now, how to represent it in code?

Now, ~~we can see~~ [→] same code for all i.e simple association, aggregation & composition.

Class A L

method() &

[Handwritten signature]

3

1

Class B L

method2() &

[Handwritten signature]

1

— 1 —

updated

Class B L

Aug 7

`btcB()` &
`a = new A();`

三

method 2 ()

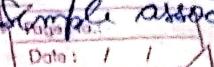
100

}) f

Main
 $B_6 = \text{new } B()$

b. method();

b. a. method 1()

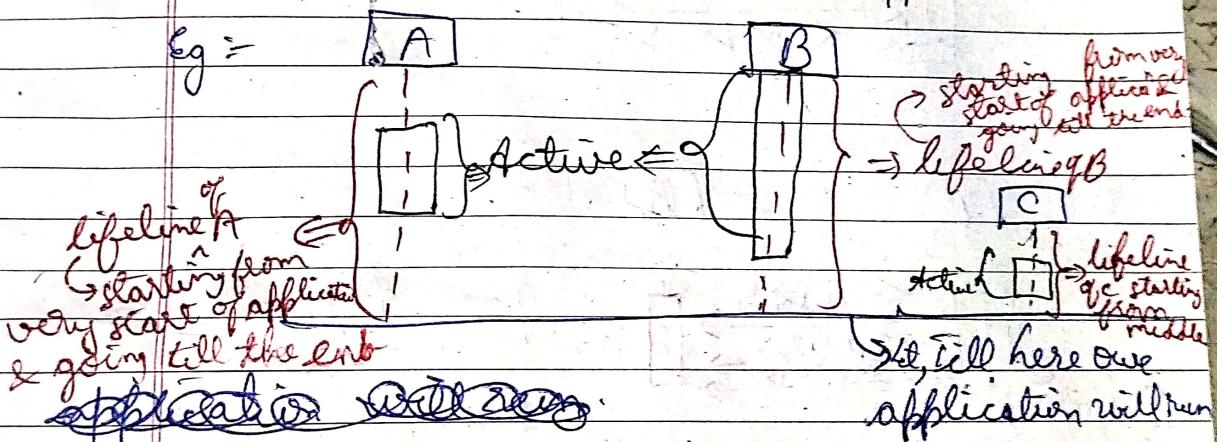


✎ ○ Basically tell about communication b/w two objects
 ✎ ○ Sequence Diagram = ○ Tell us about
 ○ when & how the interaction b/w two objects will take place
 ○ Page No.: 1 / 1
 ○ Do not care about
 the structure of class.
Components of sequence diagram
Object declare = A → B
Lifetime = Shows object A - B - C...
Activation bar = Active

(i) Object declare = [A] [B] [C] ...

(ii) Lifetime = Shows ^{two} ~~one~~ object A - B - C exist
 Are ga hamare ek paise - ke paise application me

Eg =



(iii) activation bar = Object ki time tak "active" rehega

means ab wo object akti hui me diverse object ke message le jay or receive kr skta hai.

(iv) How messages are sent =

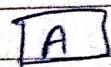
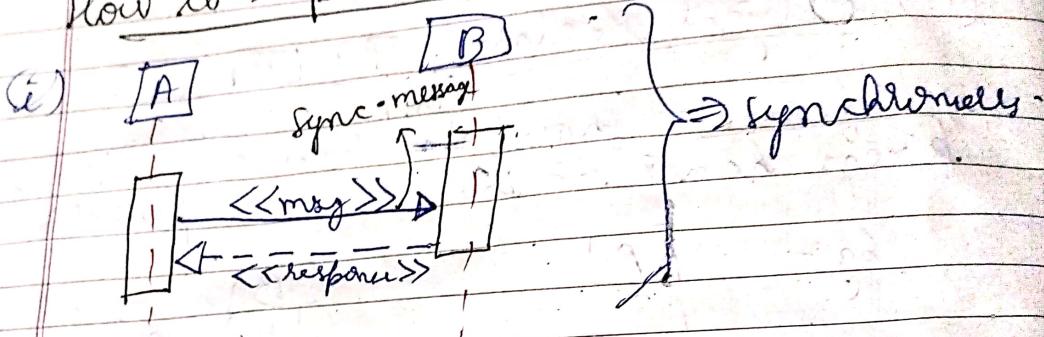
Messages are of two types =

(i) Synchronous message = It's a type of message in which we can send a message to the object & wait for its response.

Note: Tab tak response nahi aa jata tab tak diverse message nahi highte hui]

(ii). Aynchronous: It's a type of message in which we send a message one after another without even waiting for the response from the object. Do you understand them?

How to represent them?



- Object but its received -
- Reflection

•Reflexen



26

is

~~76~~ W

(i)

(ii)

卷之三

Types of messages

(i) Create message

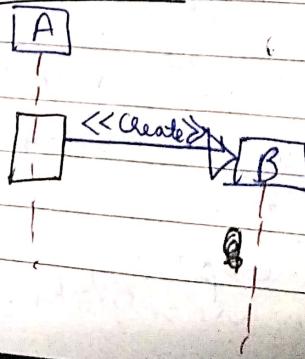
Message to

Creating an object

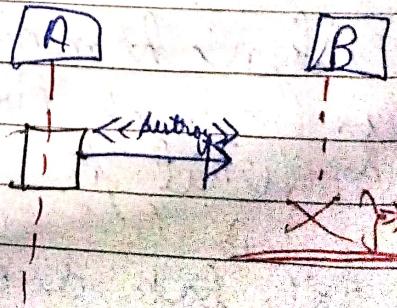
(ii) Destroy message

① Message to destroy
~~any witness~~

Representation =



○ Representation :-



(iii) Lost message

Object A sends a message but it gets lost & didn't receive to object B

Representation =

TA

B → lifetime [B]

↓ B gets

terminated before it

received a message

↓ in this

we'll call that the message is lost

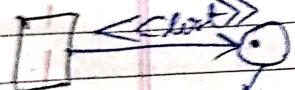
(iv) Found message

A message is received

by an object but who sent it, the object don't know

Representation =

[A]



lost message

↓ in this

→ Jai A ka message mila

B se to B ka

activation bar

khtm hogya

to 'A' ka response

leno ke liye

koi hai henti

available

How to draw Sequence diagram?

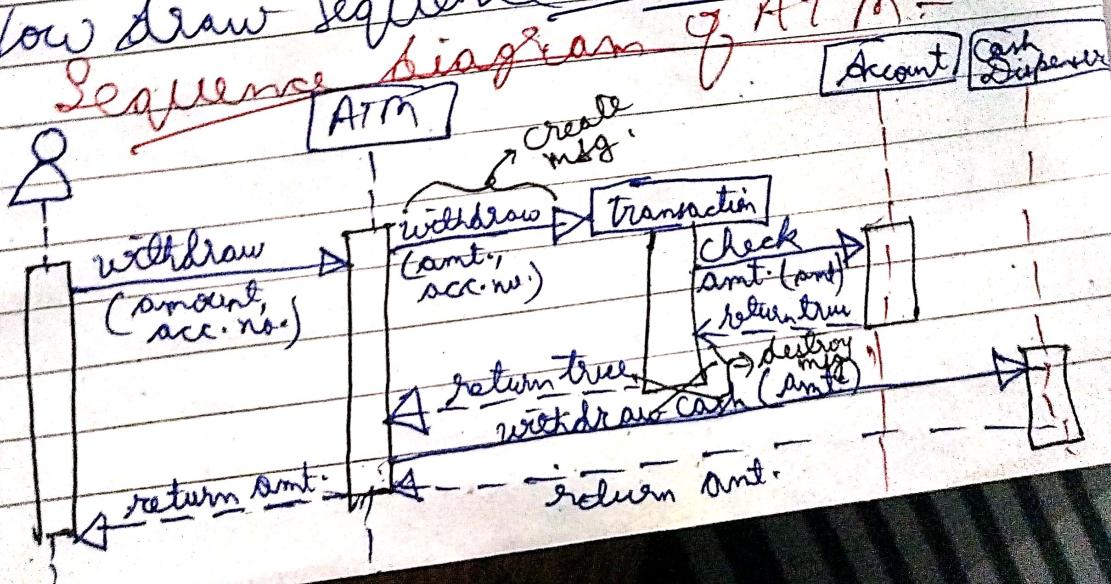
Workflow -

(i) Determine the use-case

(ii) Identify the objects

(iii) Now draw sequence diagram.

Sequence diagram of ATM:



Some other terms in sequence diagrams :-

(i) alt : (if- else) block
↳ least used

Page No.:
Date: 1/1/1

(ii) option : if block

(iii) loop : for / while