02 Objects

Introduction to OOA OOD and UML

2022 Spring

College of Information Science and Engineering

Ritsumeikan University

Yu YAN

Outline

- Types of UML Diagrams
- The Definition of an Object
- Depiciting an Object
- Encapsulation
- Associations and Aggregations
- Graphs and Trees
- Navigable Links
- Summary and Class Vocabularies
- Exercise 02

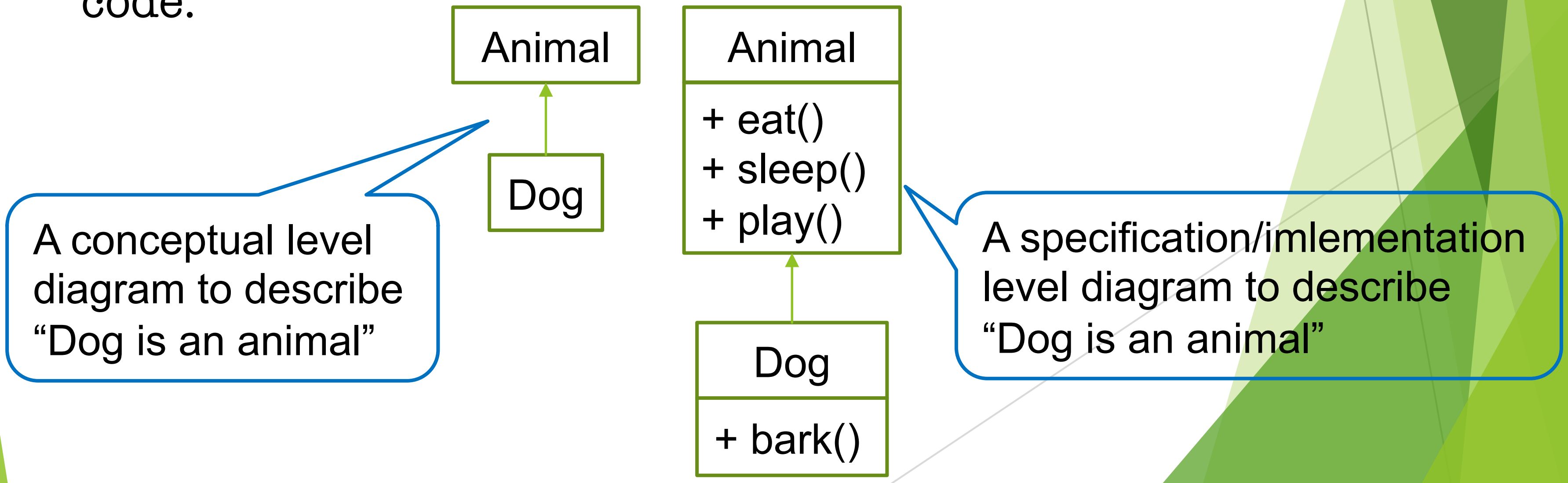
Unified Modeling Language (UML)

- The UML is a graphical notation for drawing diagrams of software concepts:

For example, a problem domain, a proposed software design, or an already completed software implementation.

Classification Method #01

- There are three different levels of UML diagrams: *Conceptual, Specification, and Implementation.*
  - Conceptual level diagrams are not strongly related to source code, but they are related to human language.
  - Specification and Implementation level diagrams have a strong connection to source code.
  - Specification level diagrams can be turned into source code.
  - Implementation level diagrams are used to describe existing source code.



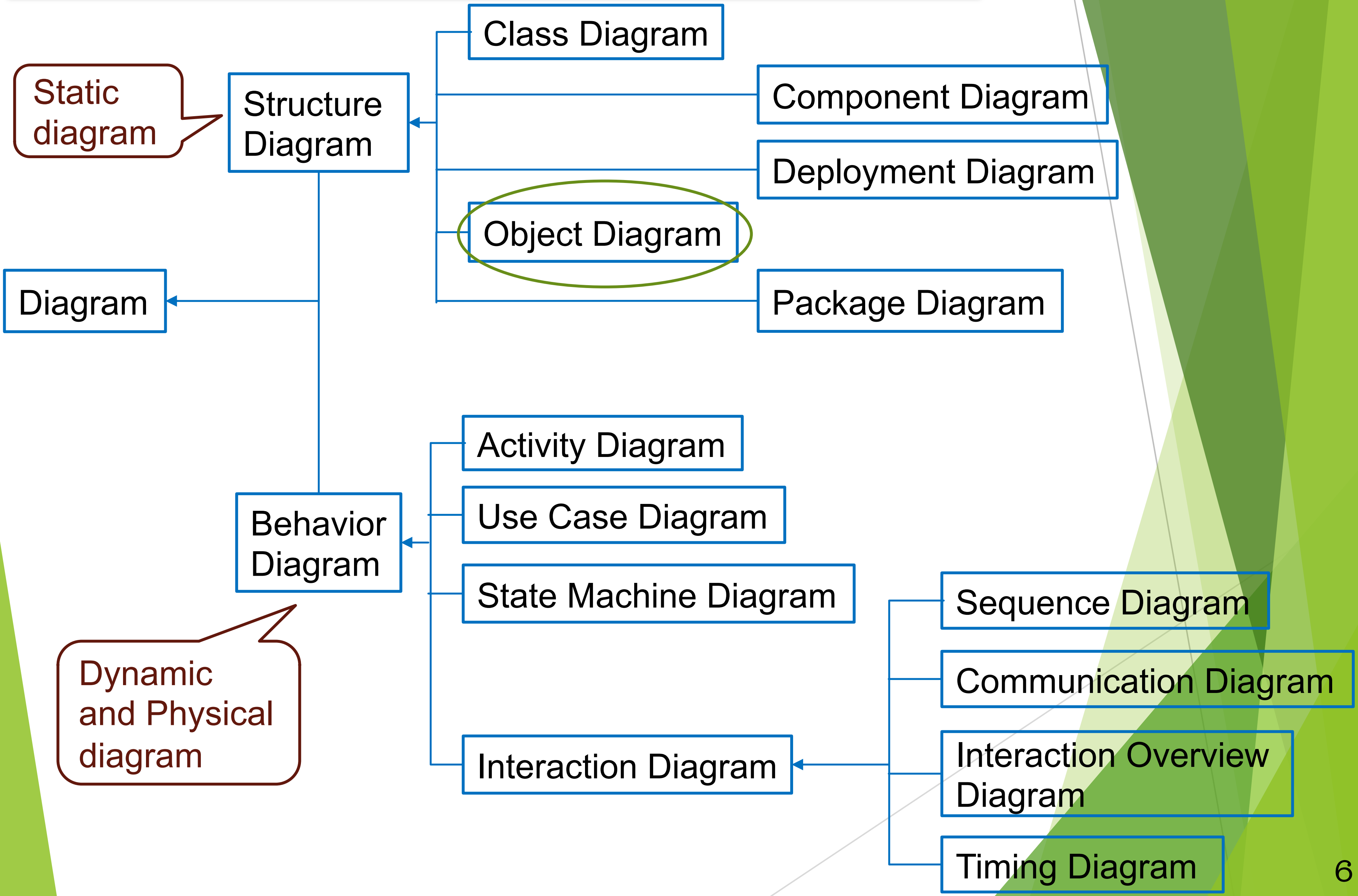
A conceptual level diagram to describe "Dog is an animal"

A specification/imlementation level diagram to describe "Dog is an animal"

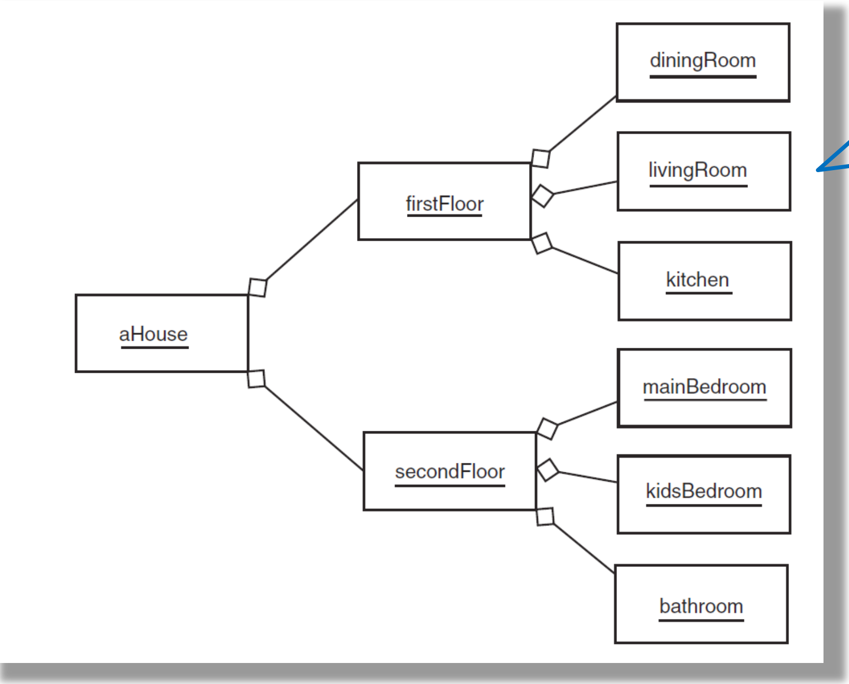
Classification Method #02

- There are three main kinds of diagrams: *Static diagrams, Dynamic diagrams and physical diagrams*
  - Static diagrams describe the unchanging logical structure of software elements by depicting classes, objects and data structures; and the relationships that exist between them.
  - Dynamic diagrams show how software entities change during execution by depicting the flow of execution, or the way entities change state.
  - Physical diagrams show the unchanging physical structure of software entities by depicting physical entities such as source files, libraries, binary files, data files, etc., and the relationships that exist between them.

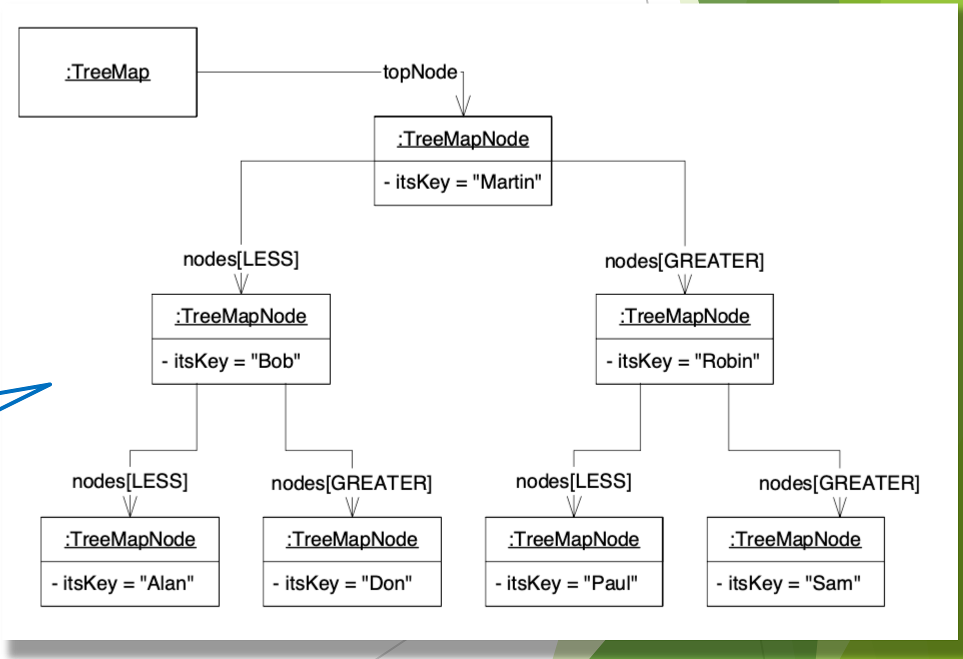
Classification Method #03



Object Diagrams



A conceptual level object diagram



A specification/implementation level object diagram

Outline

- Types of UML Diagrams
- The Definition of an Object
- Depiciting an Object
- Encapsulation
- Associations and Aggregations
- Graphs and Trees
- Navigable Links
- Summary and Class Vocabularies
- Exercise 02

Why Objects?

- Objects are easier for people to understand:

  Objects are derived from the business

- Specialists can communicate better:

  Everyone is dealing with the same concepts and notations

- Data and processes are not artificially separated:

  Data and precesses are kept together in small, easy -to-manage packeges

- Code can be resued more easily:

  Objects are like the pieces in a jigsaw puzzle: if one piece is changed, it might affect a few pieces next to it, but the rest of the puzzle will remain intact.

- Object orientation is mature and well proven:

  Applying objects in such areas as software, databases and networks is now well understood.

Objects in Real World

- DEFINITION: Object is an entity (a thing), which has its own identity, and is characterized by state/attribute and behavior.
  - An object can be not alive: a car, a building, a table, a bank acount…
  - Or alive: a dog, a cat, a person…
- State/Attribute is a collection of object's current characteristics:
  - Person's name, age, weight, height, nationality, …
- Behavior is a collection of object's changing characteristics in time:
  - Changing of human posture, start dancing, stop laughing…

What about the lifeless object? Passive

Objects in Software Engineering

- DEFINITION: Software object is a model of a real world object, which has its own identity, and is characterized by state/attribute and behavior.
  - A model is a representation of a problem domain or a proposed solution that allows us to talk, or reason, about the real thing.
- State/Attribute: Software object state is commonly modeled and called as "attributes", "fields" or "variables".
- Behavior: Software object behavior is commonly modeled and called as "operations", "methods" or "functions"

- We only model the attributes and behavior we are interested in.
- Bank custormer model needs age, gender, salary, but does not need customer's favorite song name.

Simplest Software Object Creation

- Suppose we will model humans with a software object named “Person”.
  - We have to think about what we are interested in.
- Suppose we are only interested in one state/attribute, which is “name”.
- In Java, the object will be create as follows:

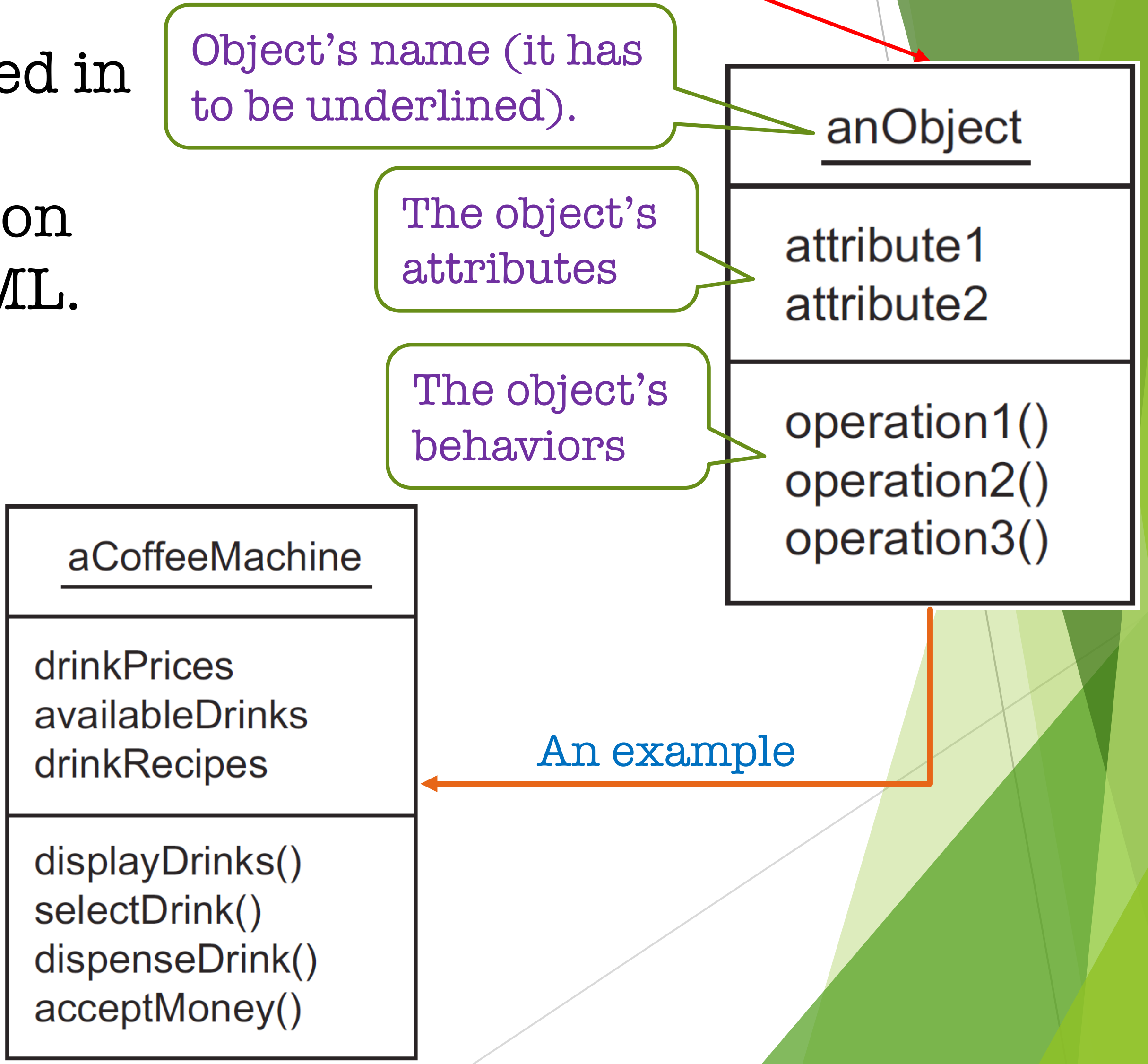
```
aPerson = new Person(“Jim”);
```

Outline

- Types of UML Diagrams
- The Definition of an Object
- Depiciting an Object
- Encapsulation
- Associations and Aggregations
- Graphs and Trees
- Navigable Links
- Summary and Class Vocabularies
- Exercise 02

Depicting Objects (1)

- An object can be described by a UML object diagram
- The parts included in an UML object diagram depend on the version of UML. However, up to 3 parts are visible.



Object's name (it has to be underlined).

anObject

The object's attributes

attribute1
attribute2

The object's behaviors

operation1()
operation2()
operation3()

An example

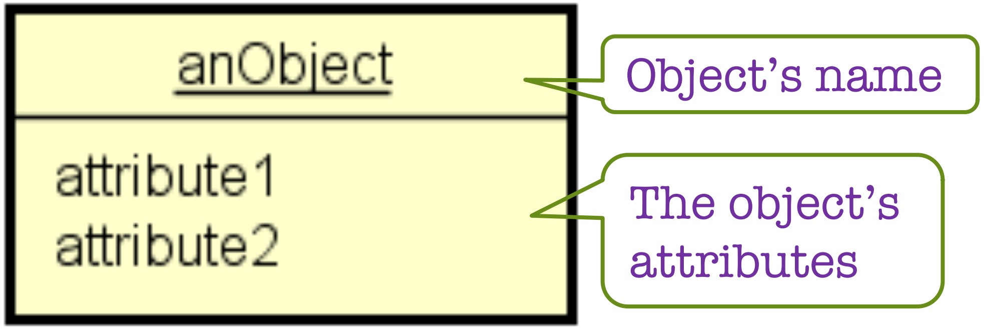
aCoffeeMachine

drinkPrices
availableDrinks
drinkRecipes

displayDrinks()
selectDrink()
dispenseDrink()
acceptMoney()

Depicting Objects (2)

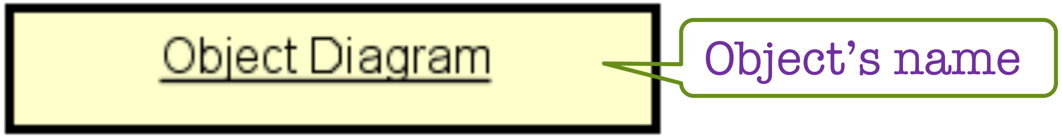
- For UML 1.5, in one UML object diagram, only two parts are visible:



Object's name

The object's attributes

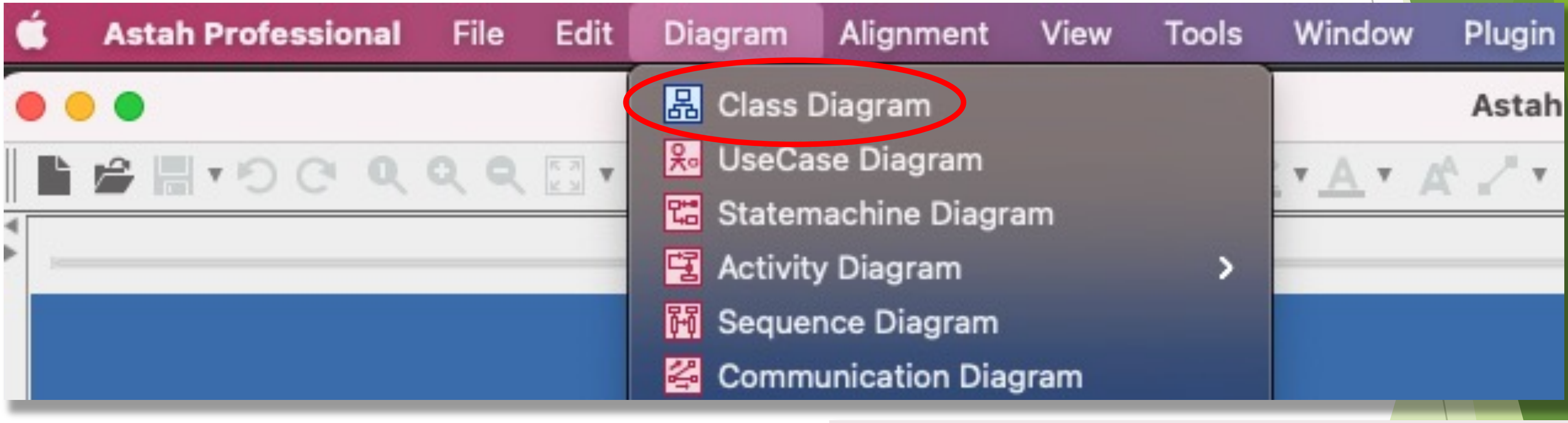
- For UML 2.0 (We will use it though the whole course), in one UML object diagram, only one part is visible:



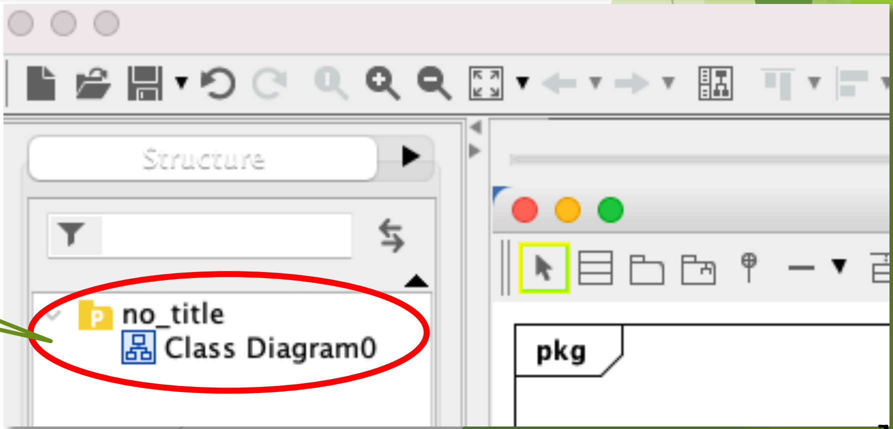
Object's name

Case Study #01 – Use Astah to Create a UML Object Diagram (1)

- Step #01: In the main menu, select "Diagram" and click "Class Diagram".

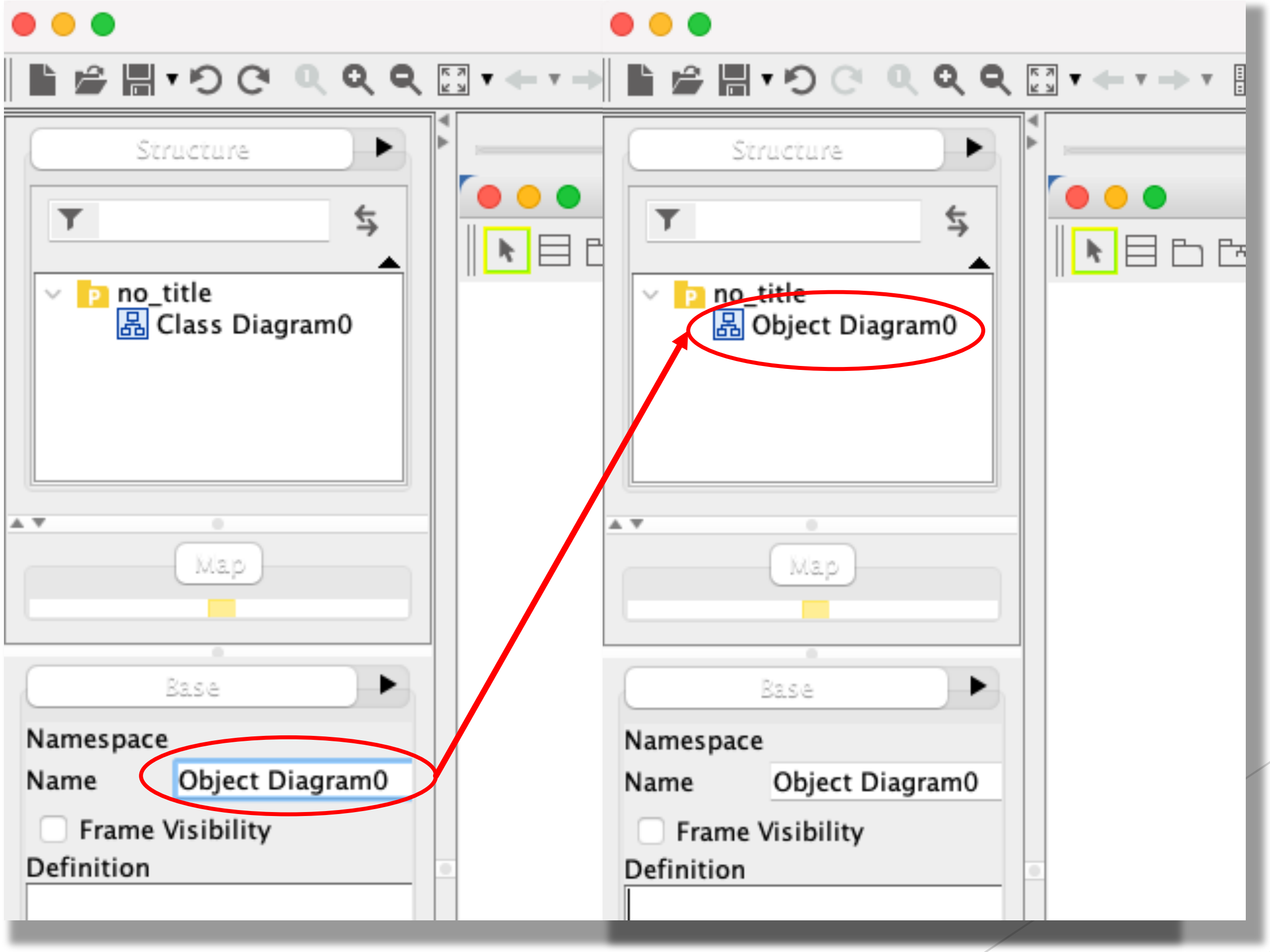


A new *class diagram* will appear



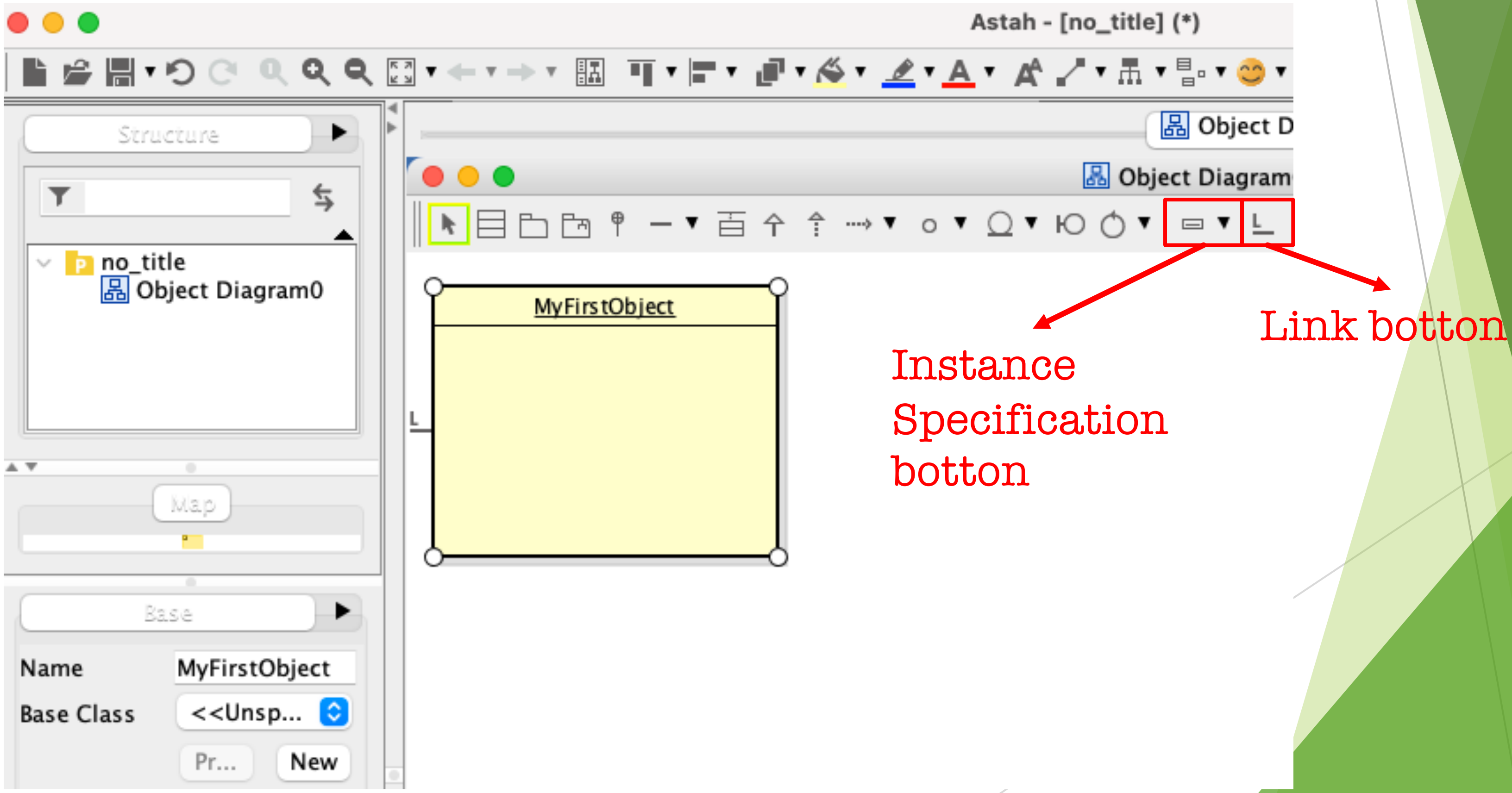
Case Study #01 – Use Astah to Create a UML Object Diagram (2)

- Step #02: Rename “Class Diagram0” to “Object Diagram0”.



Case Study #01 – Use Astah to Create a UML Object Diagram (3)

- Step #03: Draw one object box by using "Instance Specification" button and "Link" botton in icon menu.

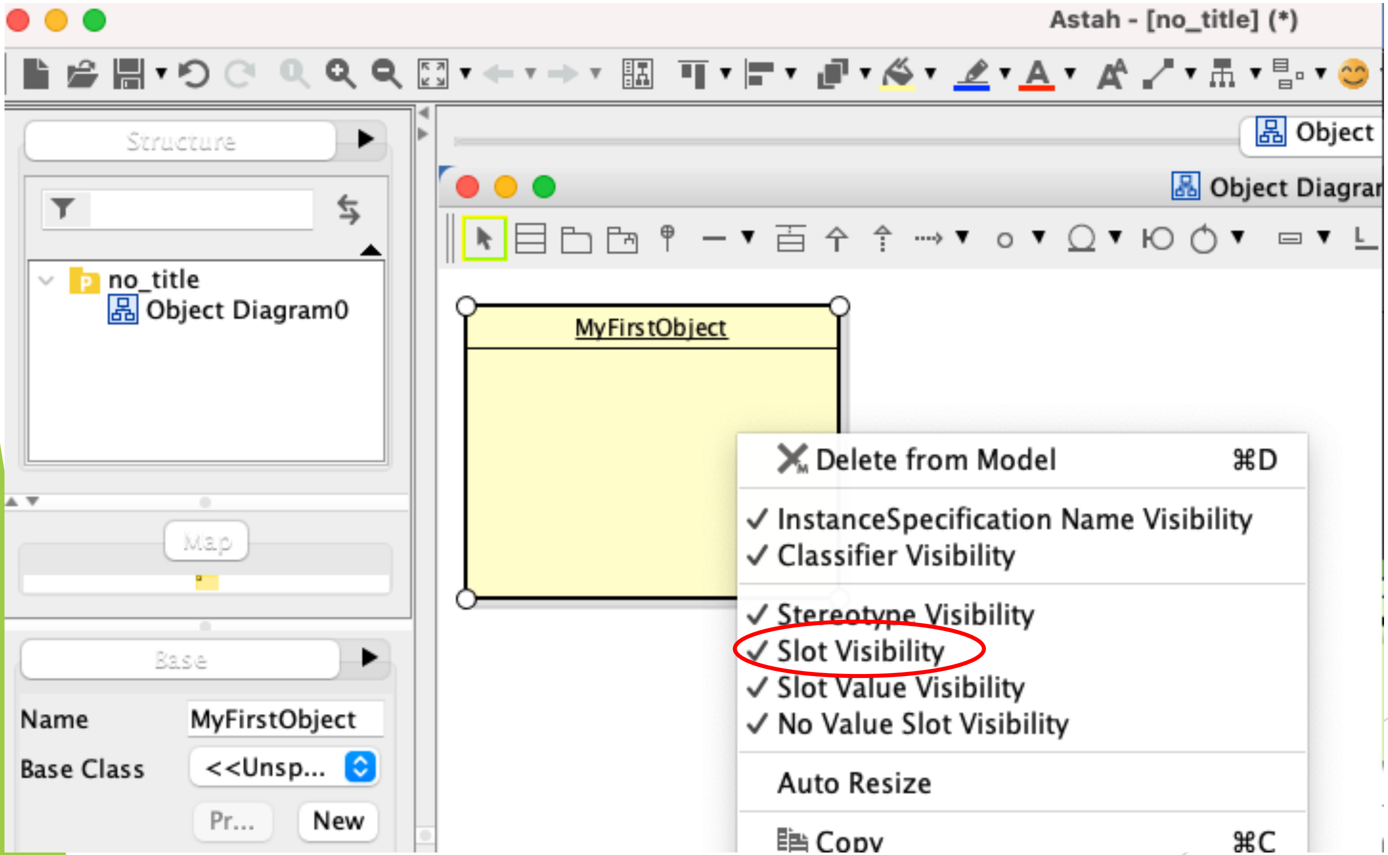


Name: MyFirstObject

Base Class: <<Unsp...

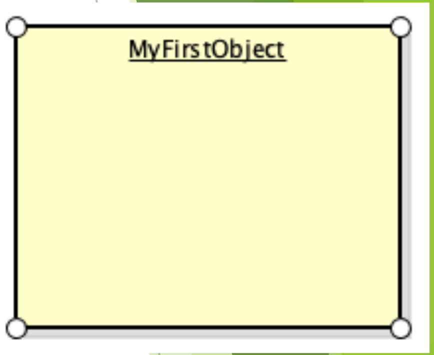
Case Study #01 – Use Astah to Create a UML Object Diagram (4)

- Step #04: Right click the object and uncheck “slot visibility”



Name: MyFirstObject

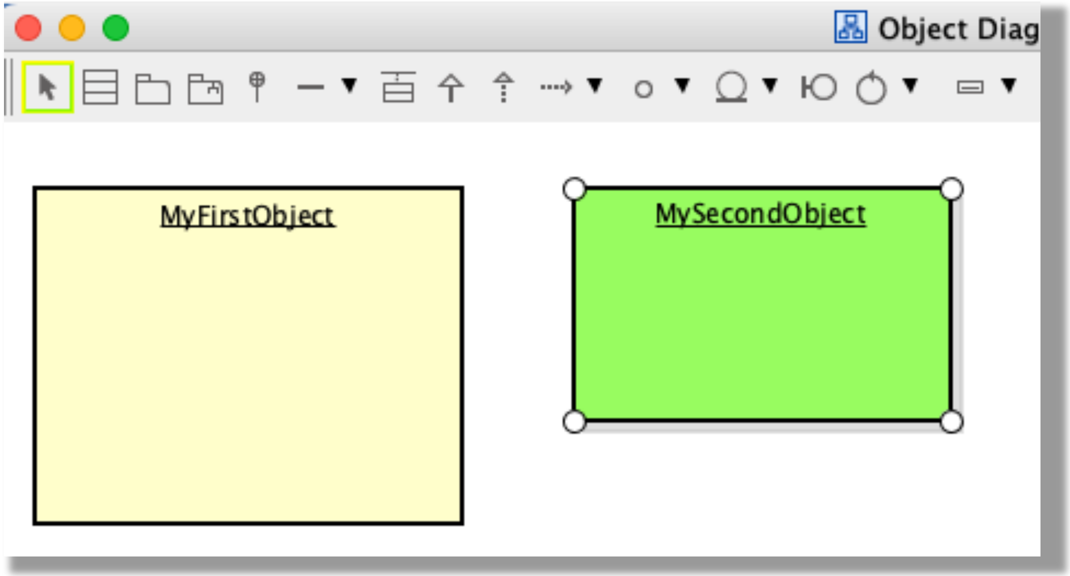
Base Class: <<Unsp...



Your first object is ready

Case Study #02 – Link Two Objects (1)

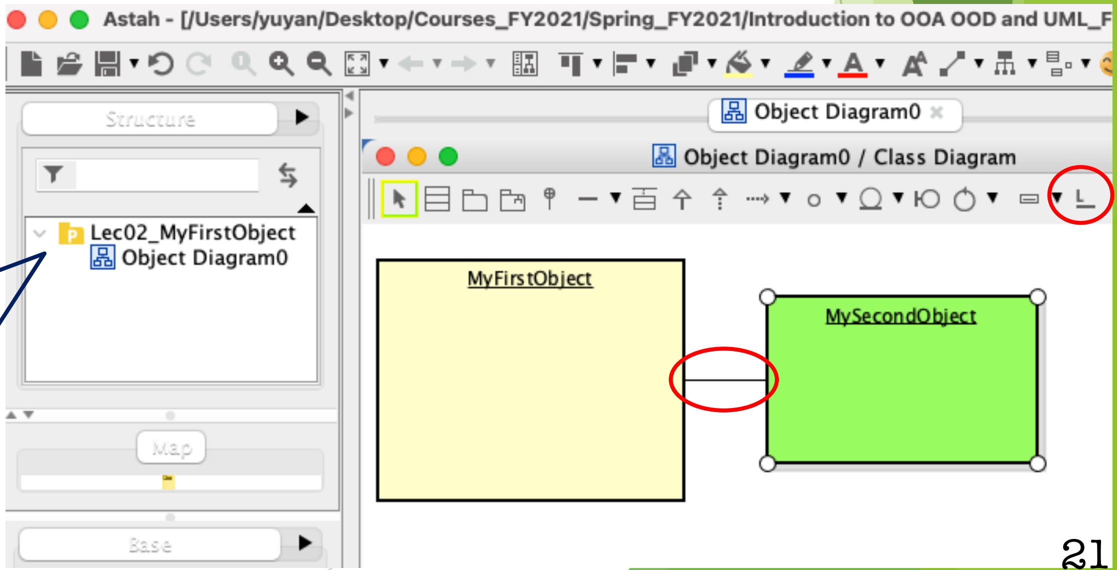
- Draw the second object and set the second object in a different color



Case Study #02 – Link Two Objects (2)

- Click on “Link” button.
- Move cursor inside any object until its border will be highlighted (e.g., by blue)
- Click-and-drag toward the second object until the border will be highlighted.
- Release the mouse – simple link will appear.
- Your first simple object diagram is ready !

- Save you diagram from Astah: in main menu: File -> Save -> save as *.astah file format.
- Save your diagram as image: in main menu: Tools -> Export Image -> Save Diagram as JPEG


Astah - [/Users/yuyan/Desktop/Courses_FY2021/Spring_FY2021/Introduction to OOA OOD and UML_F
Structure
Lec02_MyFirstObject
Object Diagram0
Map
Base
Object Diagram0
Object Diagram0 / Class Diagram
MyFirstObject
MySecondObject

Outline

- Types of UML Diagrams
- The Definition of an Object
- Depiciting an Object
- Encapsulation
- Associations and Aggregations
- Graphs and Trees
- Navigable Links
- Summary and Class Vocabularies
- Exercise 02

What is Encapsulation ?

- Encapsulation refers to an object hiding of its attributes (States) behind its operations (Behaviors).
  - Hidden attributes are called private.
- In object-oriented programming, attributes should not be accessed directly, but rather via operations
  - For example: if attribute is "color", then operation like "GetColor()" should be prepared.

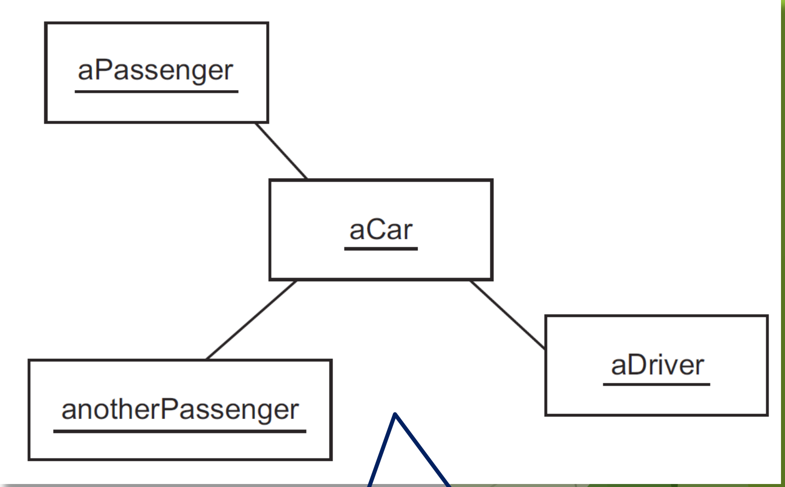
Why do we need Encapsulation?
In order to keep simplicity and locality.

Outline

- Types of UML Diagrams
- The Definition of an Object
- Depiciting an Object
- Encapsulation
- Associations and Aggregations
- Graphs and Trees
- Navigable Links
- Summary and Class Vocabularies
- Exercise 02

Association

- By connecting objects, we are able to navigate around to find extra information and behavior.
- When we are modeling with objects, association and aggregation are used to connect objects.
- Association is a weak form of conncetion:
  - Several objects may be part of a group but they are not completely dependent on each other.
  - All objects have their own life cycle an there is no owner.
  - Association links are depicted by lines.



We can remove one object and the whole won't be destroyed.

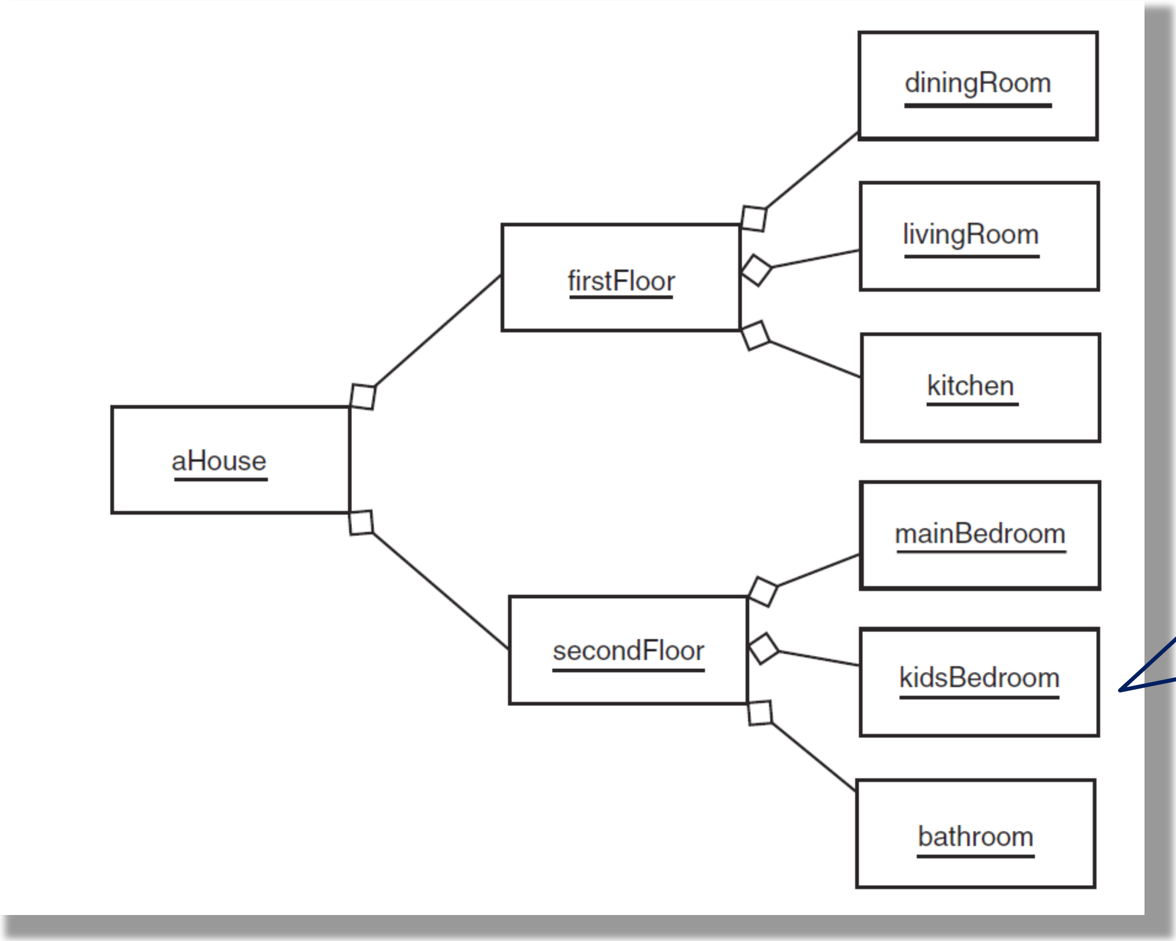
What happens if we remove one object?

Aggregation (1)

- Aggregation is a strong form of connection:
  - All objects have their own lifecycle, but there is ownership, and child objects cannot belong to another parent object
- Aggregation means putting objects together to make a bigger object
  - Manufacturing items usually form aggregations
- Aggregations form a part-whole hierarchy

Aggregation (2)

- Aggregations are depicted by lines with diamonds



What happens if we remove one object?

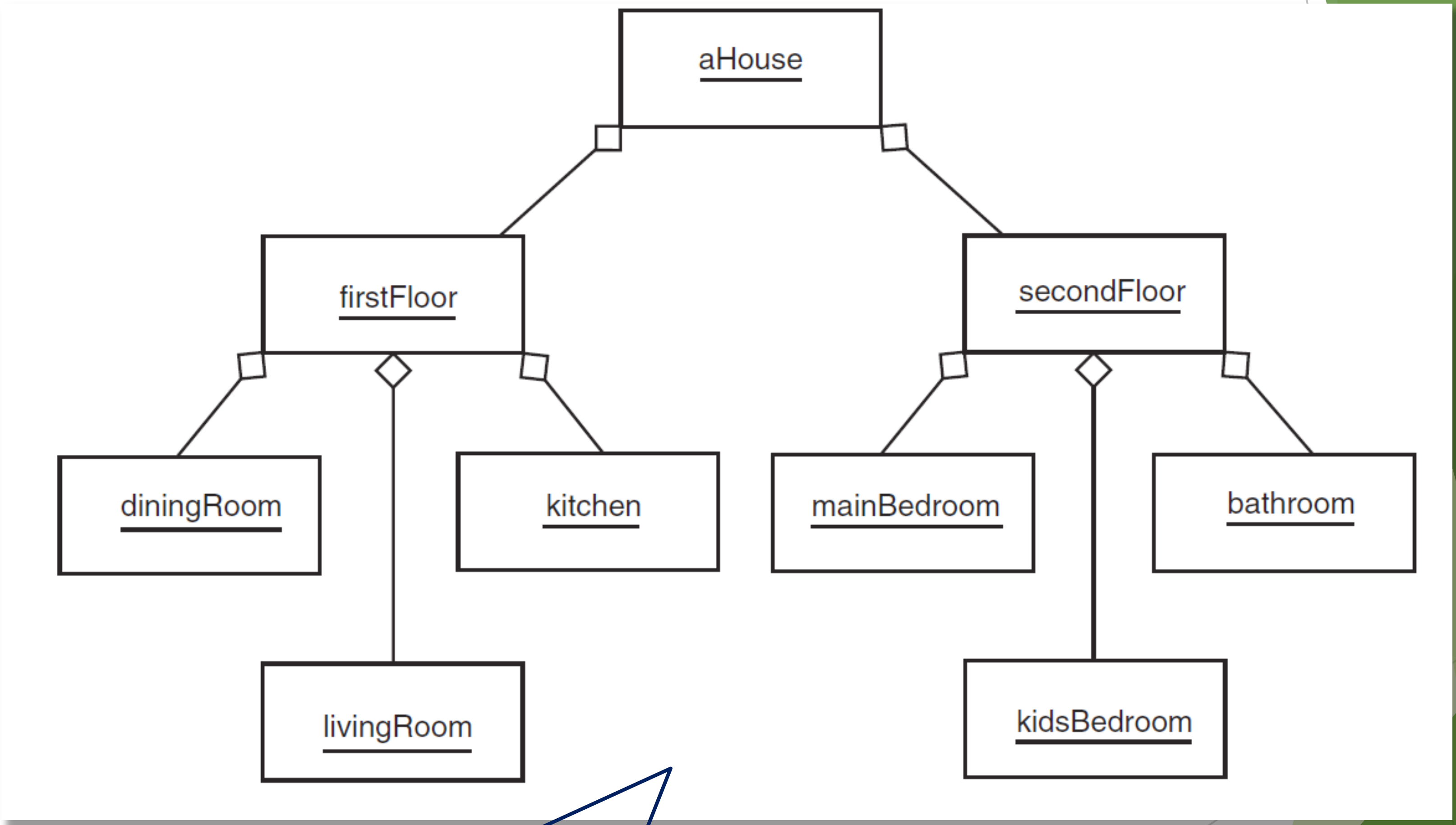
We cannot remove one object without destruction the system as a whole

Outline

- Types of UML Diagrams
- The Definition of an Object
- Depiciting an Object
- Encapsulation
- Associations and Aggregations
- Graphs and Trees
- Navigable Links
- Summary and Class Vocabularies
- Exercise 02

Trees

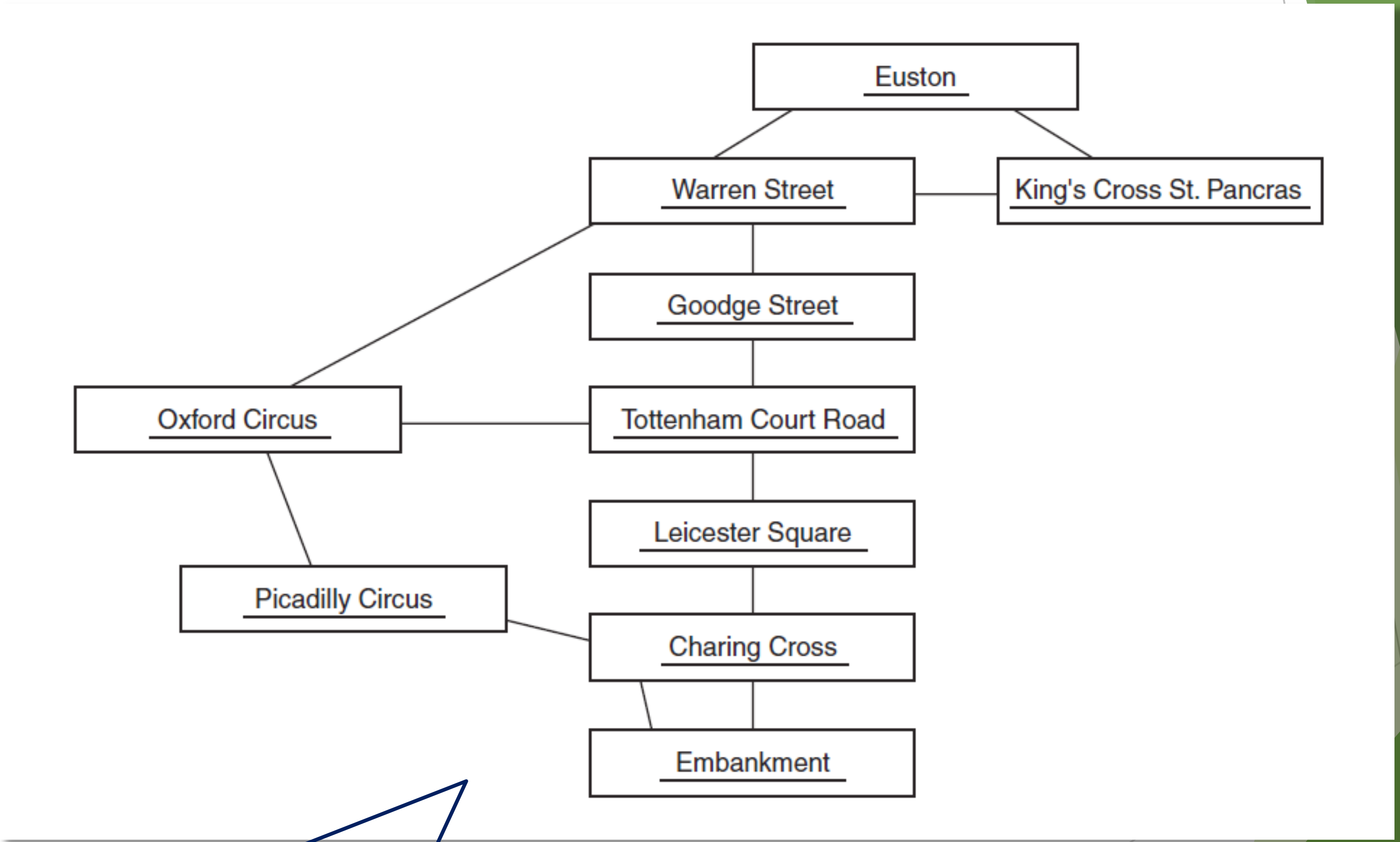
- Tree is another form of hierarchy



Aggregation as an upside–down tree

Graph

- A graph is an arbitrary set of connections among objects



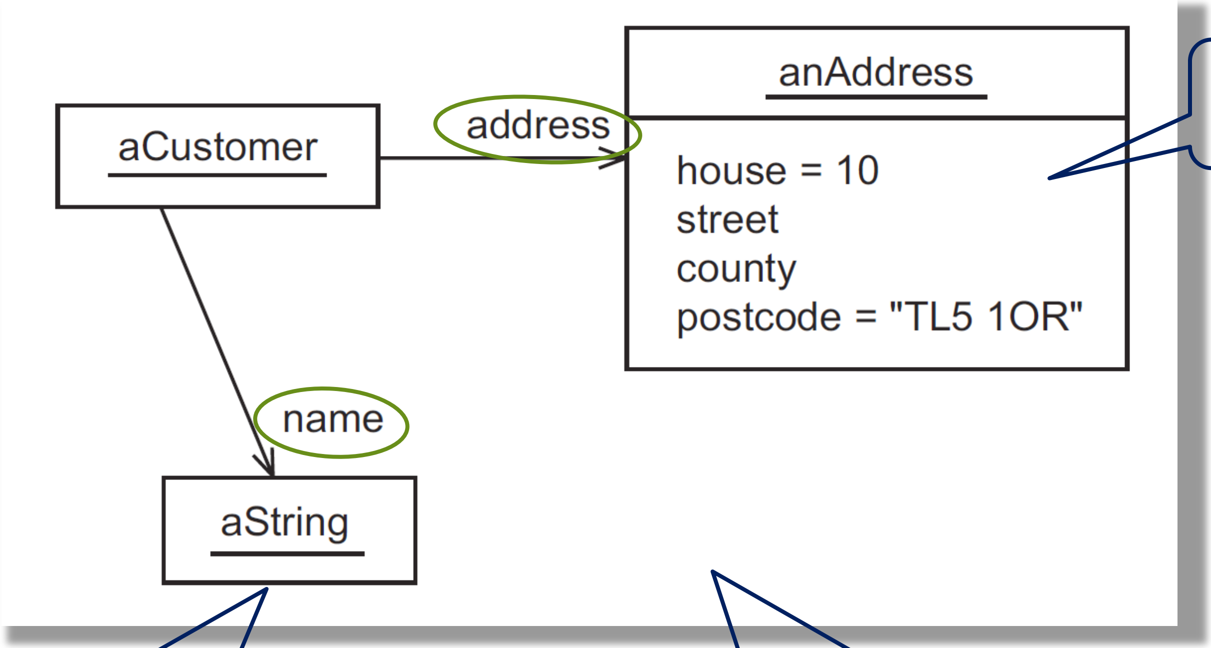
Objects in an association often form a graph

Outline

- Types of UML Diagrams
- The Definition of an Object
- Depciting an Object
- Encapsulation
- Associations and Aggregations
- Graphs and Trees
- Navigable Links
- Summary and Class Vocabularies
- Exercise 02

Links

- **Each** link can be thought of as an attribute, it connects two objects



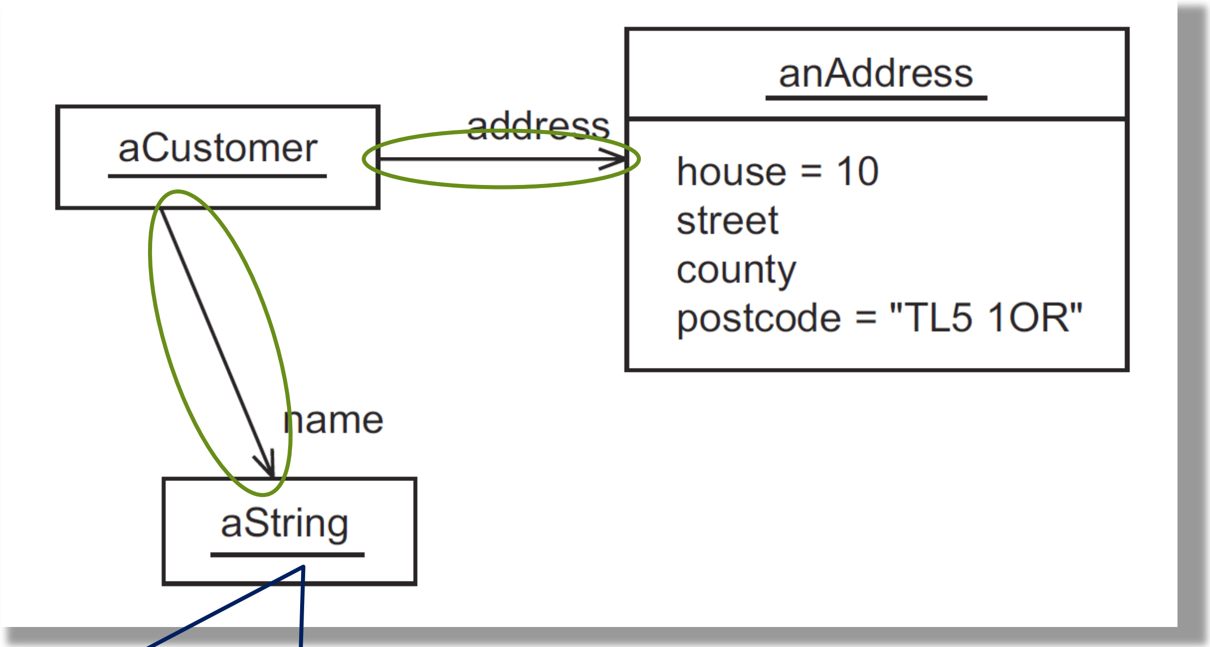
Link *address* connects to anA*ddress* object

Link *name* connects to aString object

Object *aCustomer* has two attributes/Links: *name* and *address*

Navigable Links

- The arrowhead indicates navigability of an object, which is the ability knowing where the other object is.



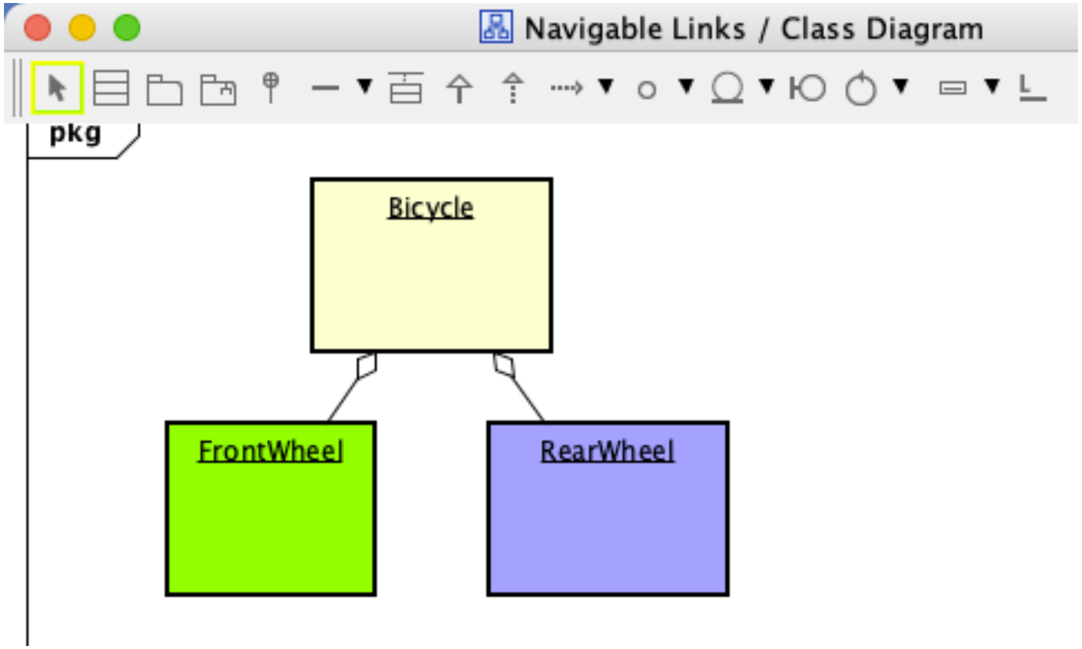
Because there's no *arrowhead* on the customer end, the implication is that Object *String* doesn't know that it is associated with *aCustormer* object

Navigatable links

- If object A knows where the object B is:
  - A link from object A to object B is navigable
  - Navigations depicted by arrowheads
  - A is often called as "source" object, and B is a target object
  - B has no knowledge about A object
  - B is invoked by A
  - Names can be assigned to links

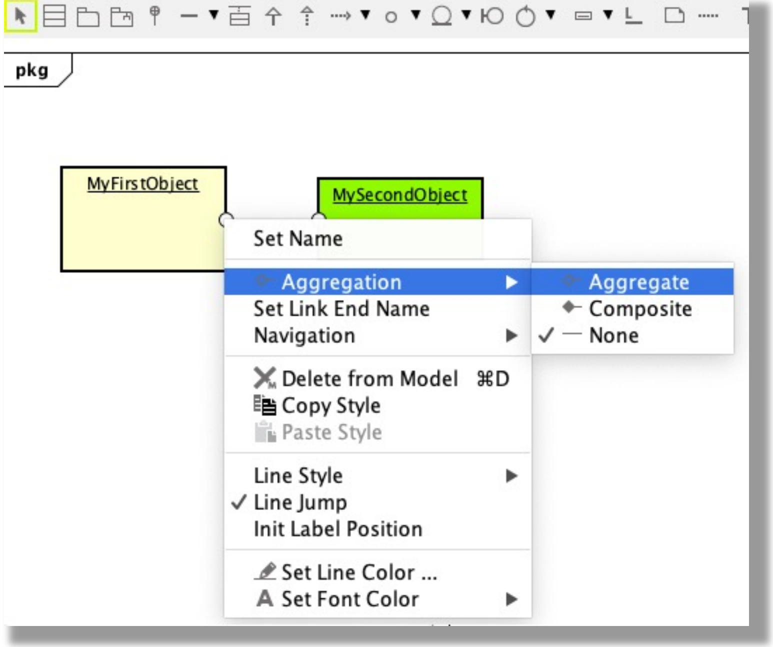
Depicting Links by Astah

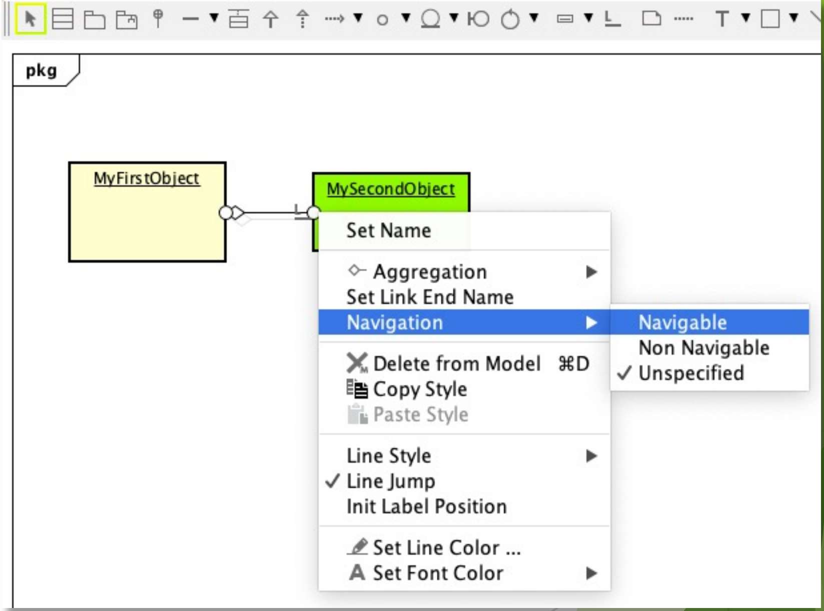
- Step #01: use "instance" and "Links" icons to create three objects ("Bicycle", "FrontWheel" and "RearWheel").
- Step #02: right click on "links" to select aggregation.
  - close to the object from which aggregation is started and depicted by white diamond.

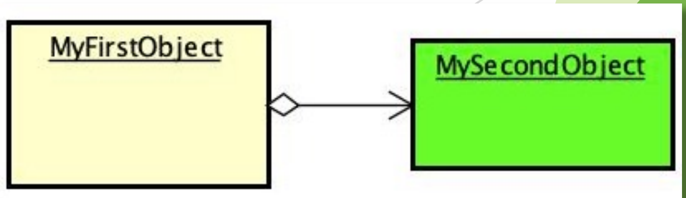


Depicting Navigable Links by Astah

- Step #01: Right click at the beginning of the link:
  - Aggregation -> Aggregate
- Step #02: Right click at the end of the link:
  - Navigation -> Navigable







Outline

- Types of UML Diagrams
- The Definition of an Object
- Depciting an Object
- Encapsulation
- Associations and Aggregations
- Graphs and Trees
- Navigable Links
- Summary and Class Vocabularies
- Exercise 02

Class Vocabularies

Objects, Attributes, Operations, Encapsulation, Association, Aggregation, Navigation

Summary

- The concepts of objects in real world and software engineering have been considered
- The types of links between objects have been studied
  - Association, aggregation and navigable links
- Initial learning of UML editor (Astah) in case of object diagram drawing was started

Outline

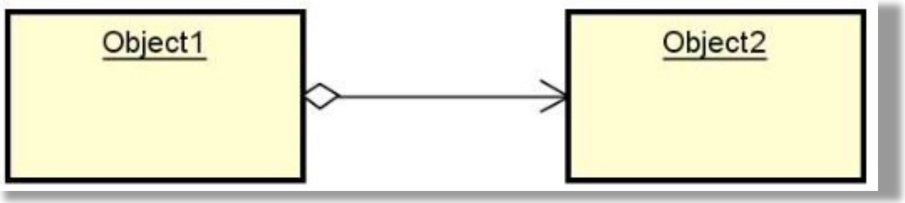
- Types of UML Diagrams
- The Definition of an Object
- Depciting an Object
- Encapsulation
- Associations and Aggregations
- Graphs and Trees
- Navigable Links
- Summary and Class Vocabularies
- Exercise 02

Exercise 02

- Deadline: **2022/04/21 (Thur.) 12:00**
- Please submit your answer file to "Exercise 02" under "Assignments" tab in Manaba +R
- Please put all of the answers in one ".pdf" file. The file name will be "**UML_Ex02_Your name.pdf**"
- The maximum points for "Exercise 02" will be **10p**
- If you put a wrong file name or wrong file format, your assignment will not be evaluated. Please be careful!

Tasks

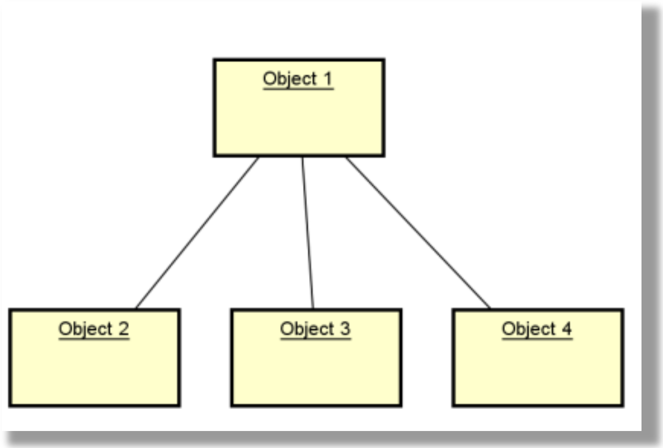
- Task #01: Consider a group of objects: the whole microwave, its magnetron and its door. Are links among the objects associations or aggregations? why? (2p)
- Task #02: Consider the following object diagram and answer the following questions



- ✓ Is the link between “Object 1” and “Object 2” aggregation? (Yes / No) (1p)
- ✓ Is “Object 1” the target object? (Yes / No) (1p)
- ✓ Is the link between “Object 1” and “Object 2” navigable? (Yes / No) (1p)

Tasks

- Task #03: Consider the following object diagram/hierarchy structure and build a similar object diagram with names ("BicycleFrame", "FrontWheel", "The whole bicycle assembly", "RearWheel") via Astah UML



- Requirements:
  - ✓ Use different colors for each block (1p)
  - ✓ Include your name into one of object names (e.g "FrontWheelJim") (1p)
  - ✓ Use link connections ("Navigable" / "Aggregated") (1p)
  - ✓ Export your diagram as a "PNG/JPEG" image and insert the image in your report (2p)