SOFTWARE ENGINEERING

Chapter 3.1 B: System Design

MOTIVATION...

Vision is the art of seeing what is invisible to others [Johnathan Swift]

System Building Levels

High Level

Requirements

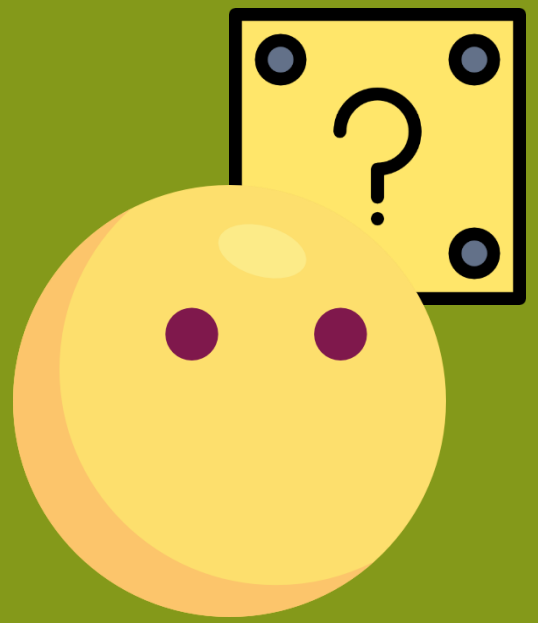
Mid Level

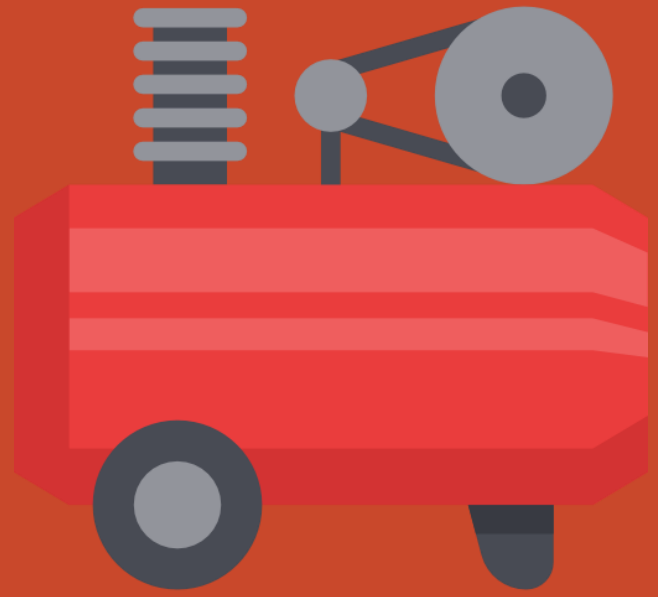
UML

Low Level

Code

What about the topic?





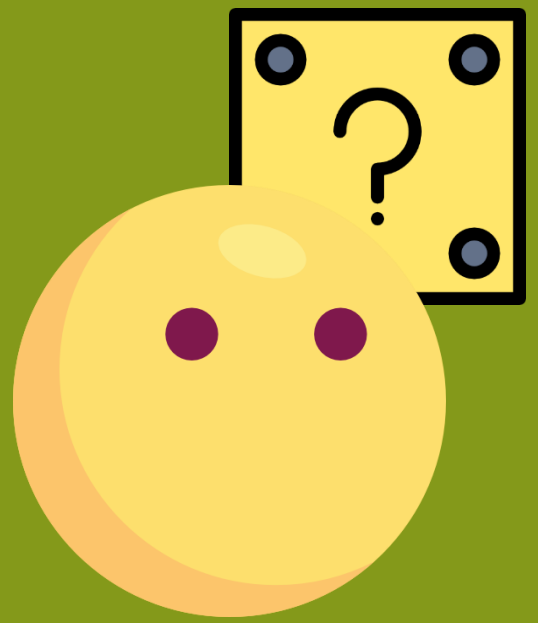
Interaction Diagrams

Describe how groups of objects collaborate in some behaviour

Interaction Diagrams

- They capture the behaviour of a simple use case.
- They don’t capture a global behaviour of the system they just capture a given scenario.

What do you mean with behaviour?



The behaviour Word!

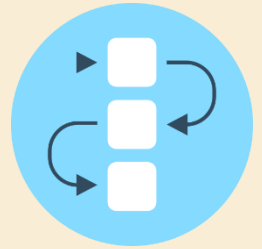
In animals and humans behaviour is defined as a reflex produced by a given stimuli or a consequence to something...

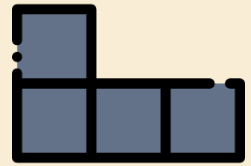
The behaviour Word!

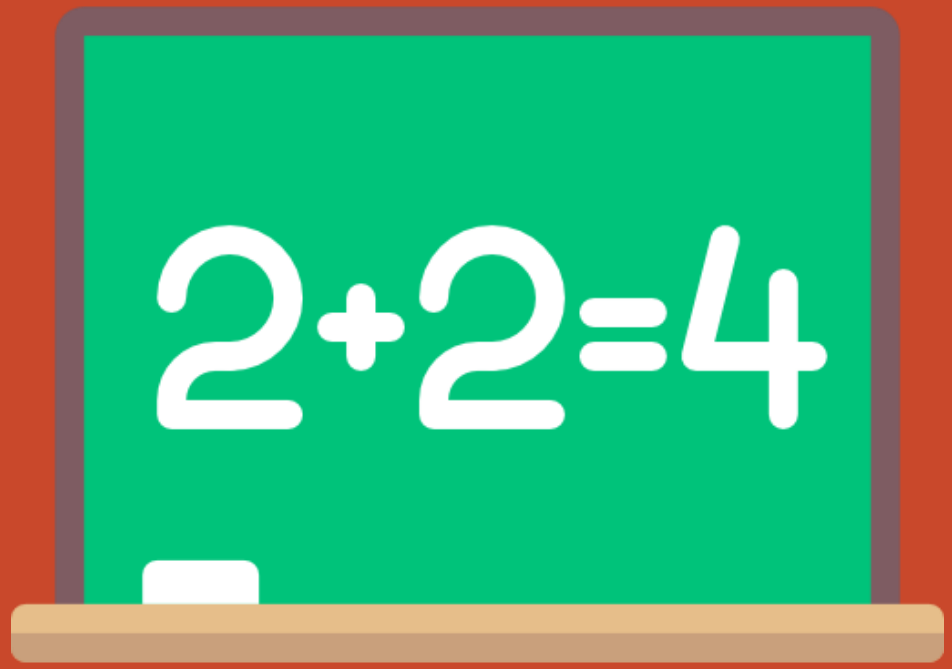
A system is a set of interconnected things in such way that they produce their own pattern of behaviour

Interaction Diagrams

- They will look at interactions between objects
- There should be one interaction diagram for use case







Sequence Diagrams

Emphasize order and concurrency of interactions

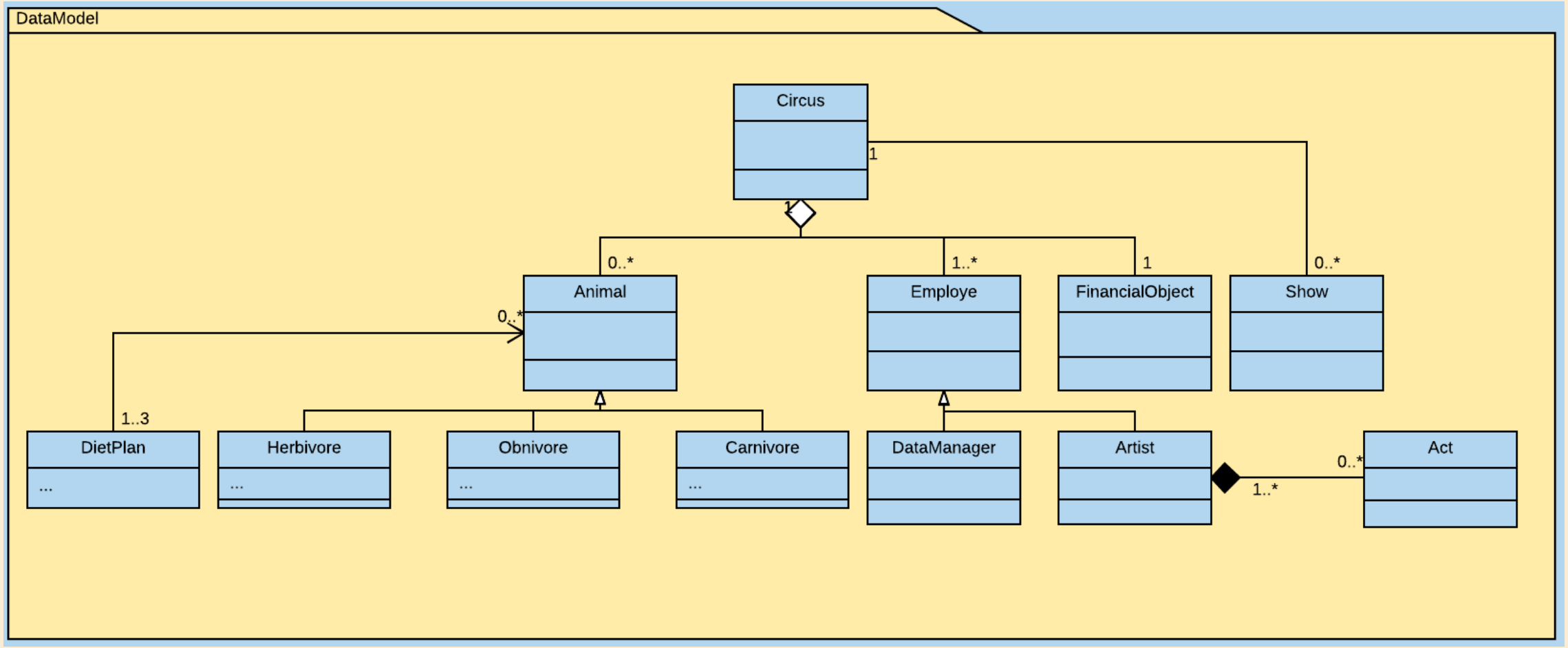
Sequence Diagrams

Emphasize the communication timing between objects.

You can only include objects who have an association between the involved classes

(the association is the main reason for communication between objects).

Let's use this!

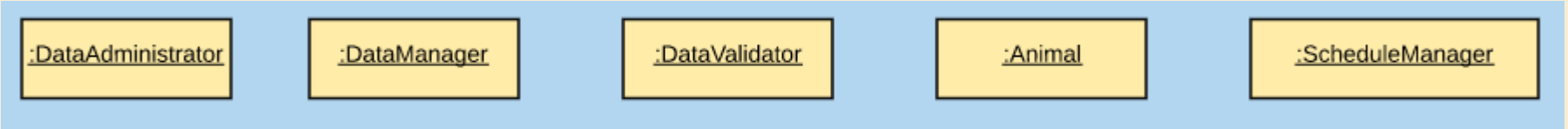


And we may also need our use case!

Use case title: Add a new Animal
Description: This use case shows how to add an animal to the database
Actors: Database, Administrator
Pre-Conditions: Administrator must be logged in, Internet connection must be detected, Server turned on and services enabled.
Post-Conditions: A new animal is added to the database
Normal Flow: 1) The system asks to re enter the admin password 2) The user enters the admin password 3) The system asks for the permission or ID of the new animal 4) The user enters the ID 5) The system displays a form asking for, type (wild, domestic), name, age, health status, diet, type of act, description of the act and a tentative Schedule for feeding, Schedule for training and for healthcare. 6) The user enters all the requested data and accepts. 7) The system sends a request for the vet, the artist and the trainer to accept the Schedule, and it registers the new animal.
Alternative Flow: 2a) The password is wrong: Return to the normal flow step 1 and count +1 for the errors. 2b) The password is wrong 3 times: Break the use case. 3a) The characters for ID are wrong: Count the error +1 and return to the step 3, if it's wrong 3 times, throw an error and break the use case.

Sequence Diagrams - Drawing

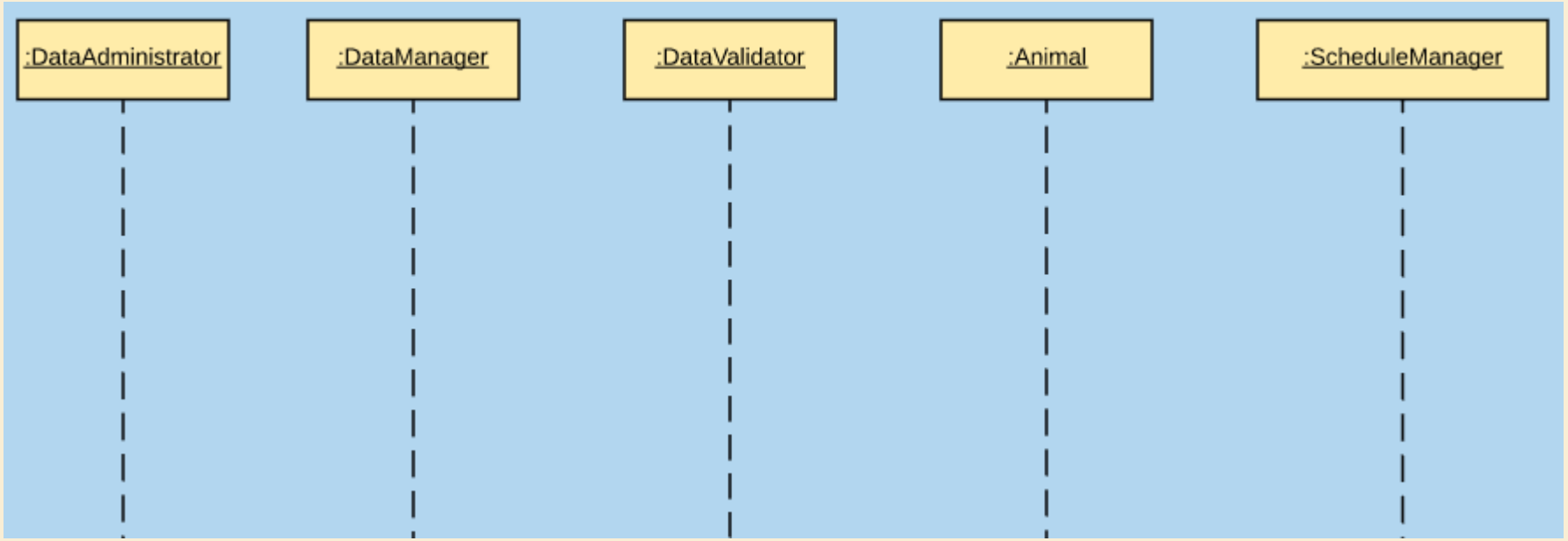
- Draw all the possible involved objects in the interaction.



You don't actually need to name the instances, like this!

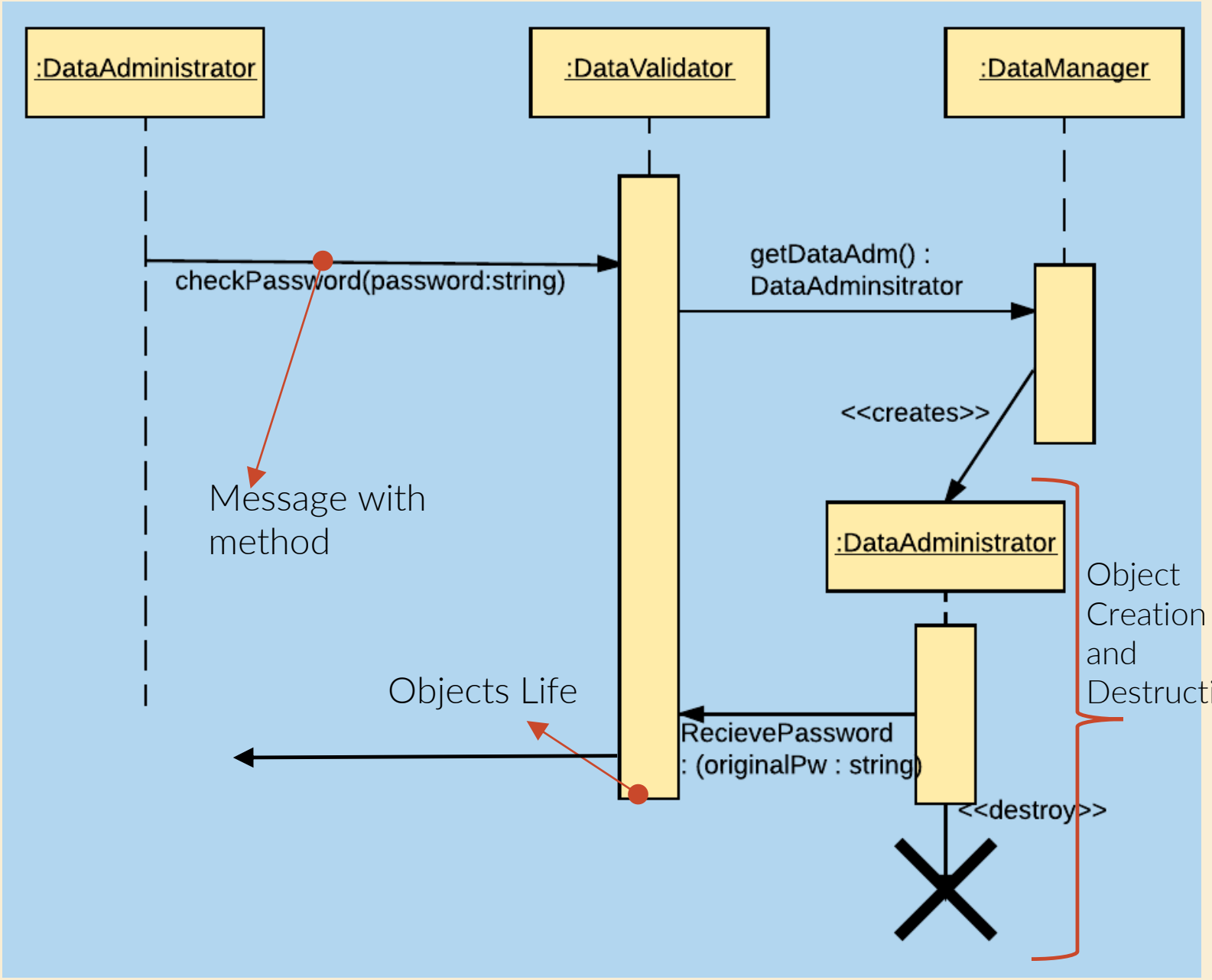
Sequence Diagrams - Drawing

- The objects are going to be ‘alive’, use that dotted line to point where the object is created and it’s life.

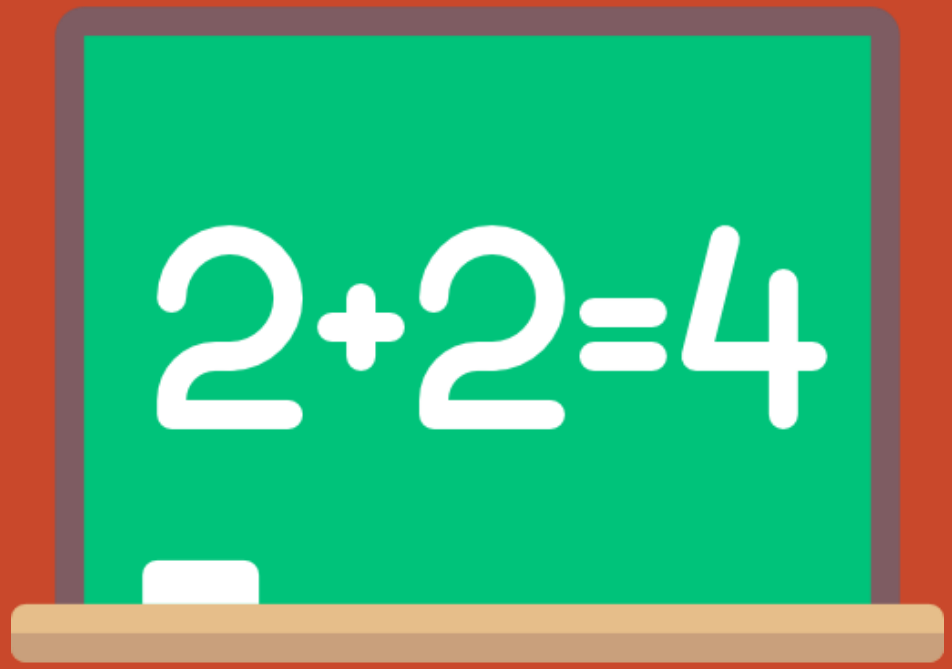


Sequence Diagrams

- The arrow is a communication (method)
- You can create temporal objects! (like :DataAministrator)



Complete it please!



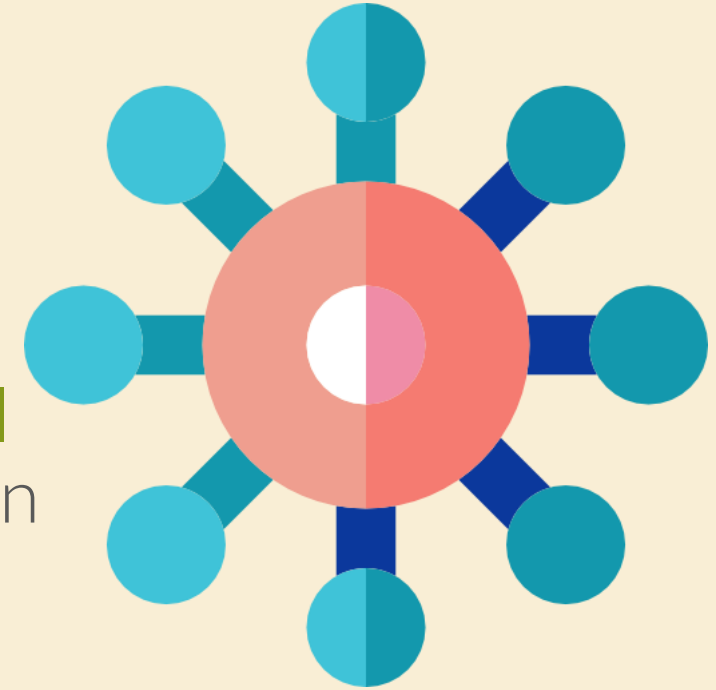
Collaboration Diagrams

Emphasize the interacting objects

Collaboration Diagrams

- They can have the same information as the sequence diagrams

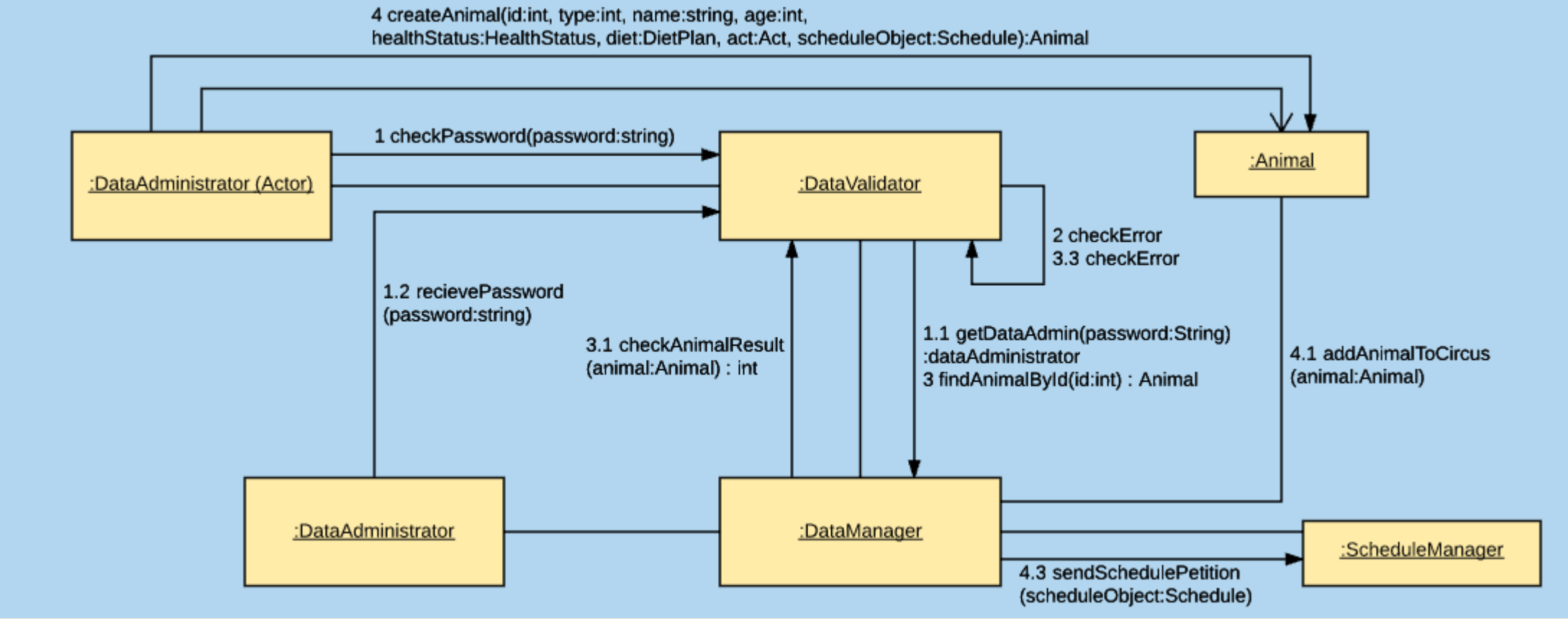
- The key difference is that they are an structural view of the interaction instead of the interaction sequences



Let’s jump straight to the example!

Collaboration Diagrams

- Interactions are numbered



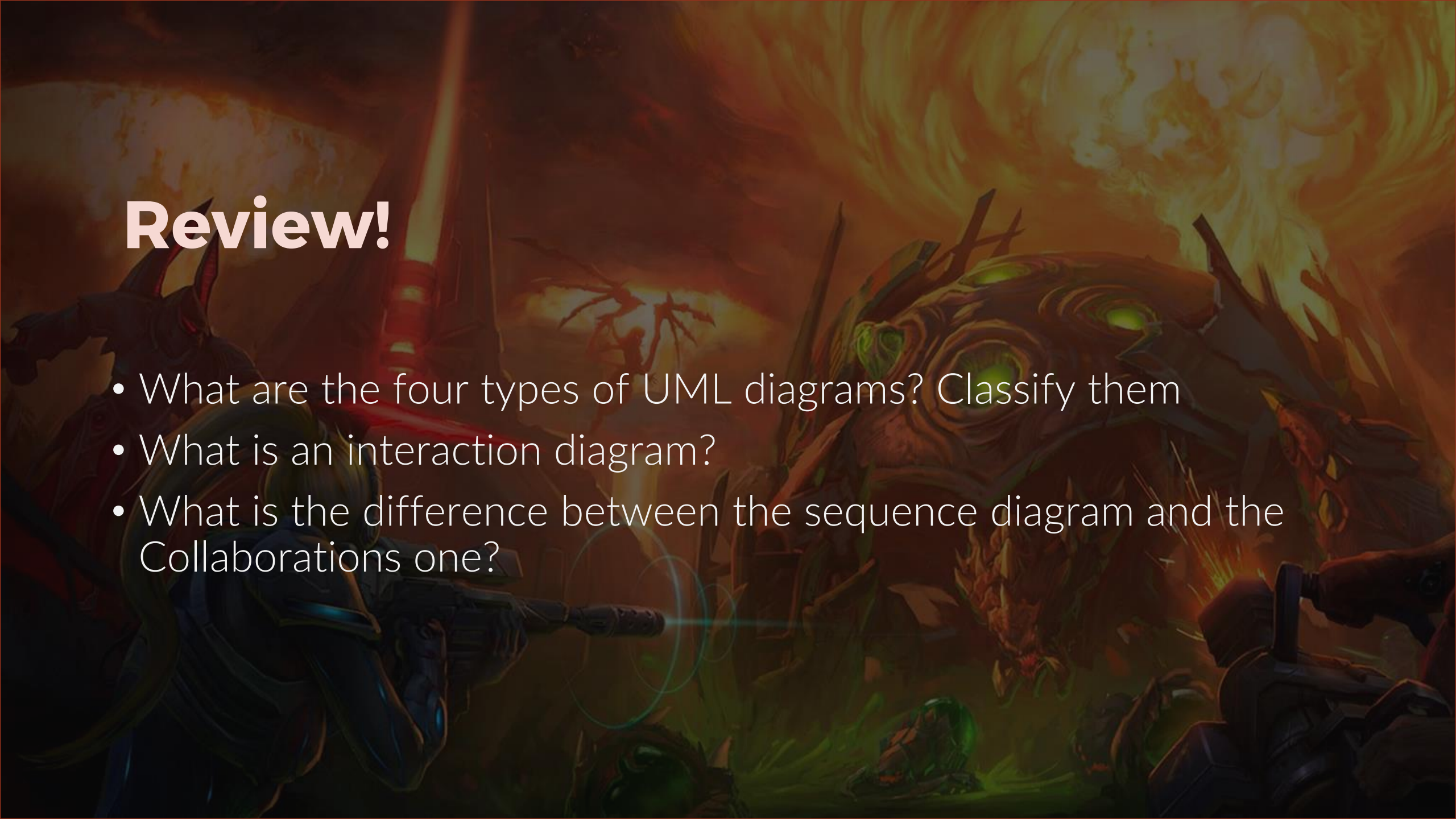
You can read the same information that you had on the sequence diagram!, the only change is that you put the relationships between classes and the communications!

Workshop time!

Remember your homework is to see by yourself these two UML diagrams as I'm gonna ask you:

1. Entity-Relationship (newer version)
2. Timing Diagram


One Shot Review



Review!

- What are the four types of UML diagrams? Classify them
- What is an interaction diagram?
- What is the difference between the sequence diagram and the Collaborations one?

References

- [SOMMERVILLE] Ian Sommervile. *Software Engineering 9th Edition*
- [SCHMIDT] Richard Schmidt. *Software Development Architecture-Driven Software Development*
- [LARMAN] Craig Larman. *Applying UML and Patterns*
- [IBM] IBM Knowledge Center – UML
- [SCHMULLER] Joseph Schmuller. *Teach yourself UML in 24 hours*
- [SCHMIDT] J.W Schmidth, F. Matthews. *Object Oriented Analysis and Design Ch 3.4 Interaction Diagrams*


Class has died... for today!