

# UML Diagrams

## Sequence Diagram

25 August 2023

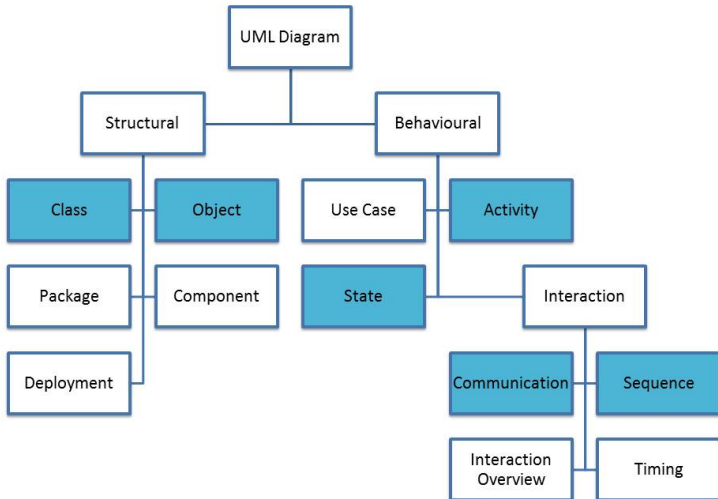
# Introduction

Notational elements

Branching

Iteration

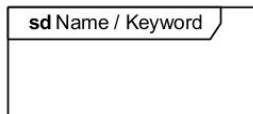
Parallel actions



## A sequence diagram:

- is used to model how objects interact with one another in terms of the messages (method calls) they pass to one another.
- emphasises the order of message execution as a reaction to some event.
- arranged interactions from top to bottom, following their order of

Sequence diagrams are drawn in *frames* - a rectangle with a heading

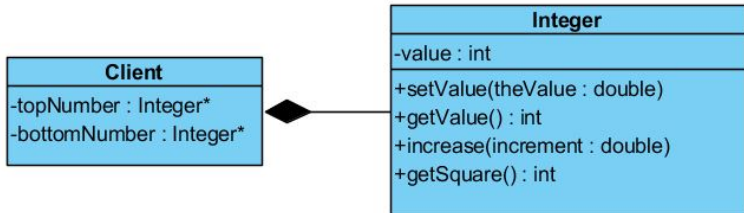


- *Name* - to name the diagram or
- *Keyword* - to indicate the scope of loop structures, conditional statements or parallel flows.

A *lifeline* represents an individual participant in the interaction (object instances).

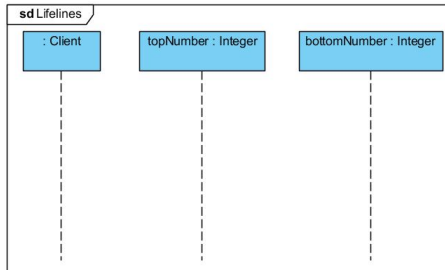


# The Integer class is a wrapper for an integer value

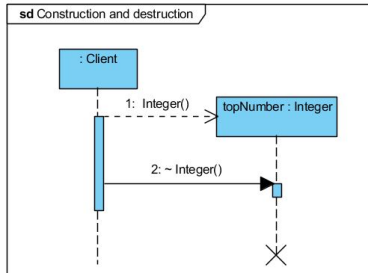


```
Integer* topNumber = new Integer();
Integer bottomNumber;
```

Objects in the system form the lifelines, the order of these lifelines is not significant.



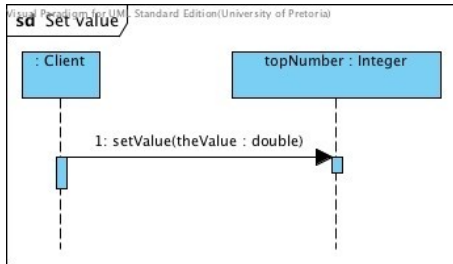
## Creation and Destruction



```
Integer* topNumber = new Integer();  
delete topNumber;
```

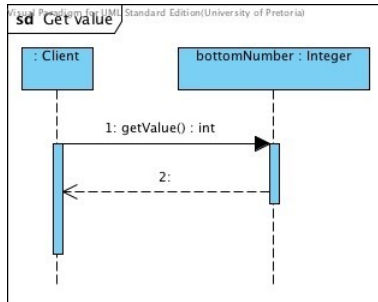


## Asynchronous - e.g. set value



```
double aValue = 2.66;  
topNumber->>setValue(aValue);
```

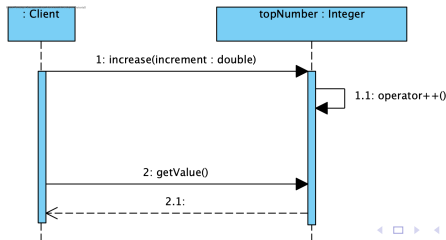
## Synchornous - e.g. get value



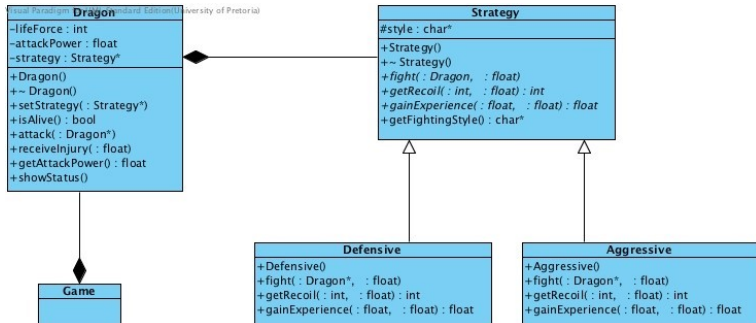
```
int aValue = bottomNumber.getValue();
```

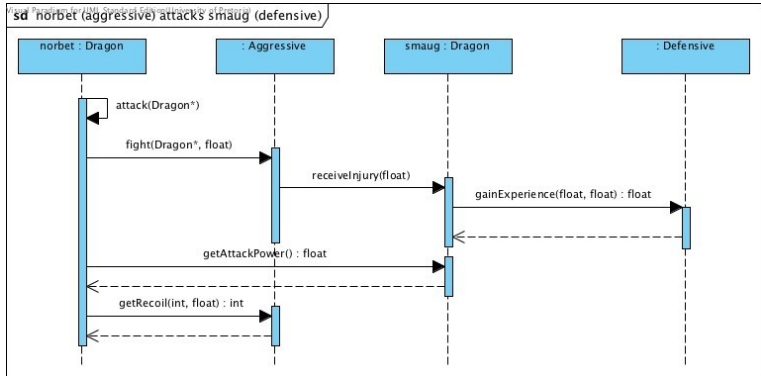
**Reflexive messages** (self message), when an object calls a method that is defined in its own class.

Assume `operator++` is a private function define in `Integer` and `increase` calls the operator increment times.

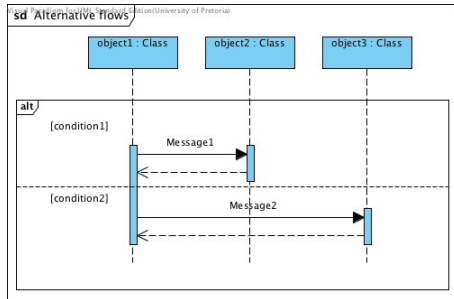


# Example

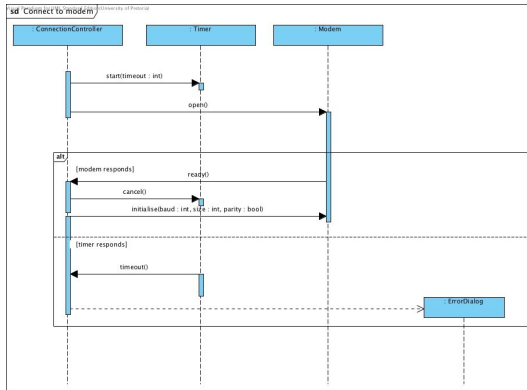




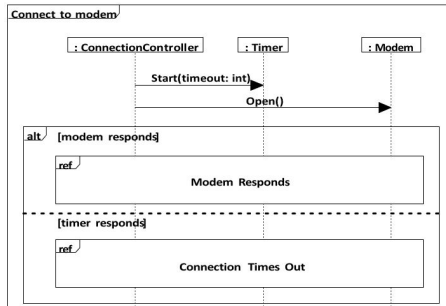
**Branching** happens when the program flow contains conditional statements.



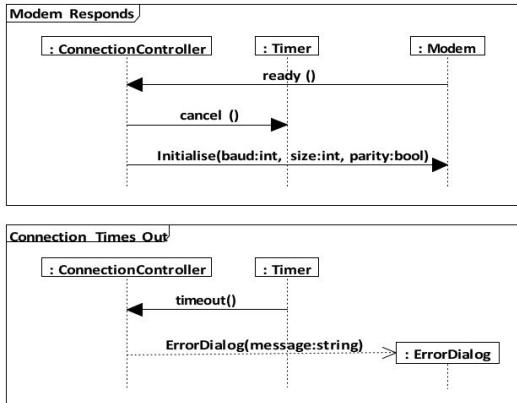
## For example, connection to a modem



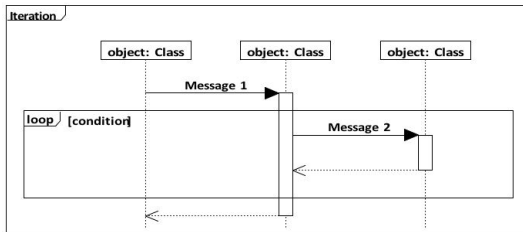
If a diagram becomes complex, it is advisable to model it in **fragments**.





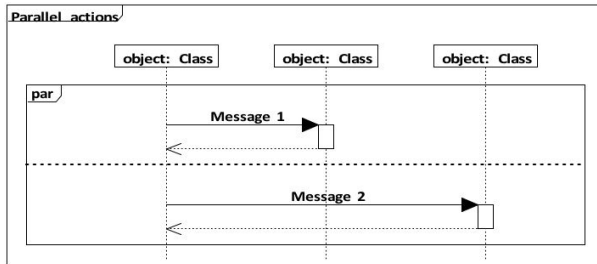


**Iteration** happens when the program flow contains looping statements.



Syntax for a loop structure

**Parallel actions** model interactions that are executed at the same time (in parallel).



Syntax for parallel actions

Introduction  
Notational elements  
Branching  
Iteration  
Parallel actions  
Parallel actions

