OOP design diagrams

Basic diagrams that we are gonna discuss here is

- use case diagram
- sequence diagram
- activity diagram
- communication diagram
- object diagram
- communication diagram
- class diagram

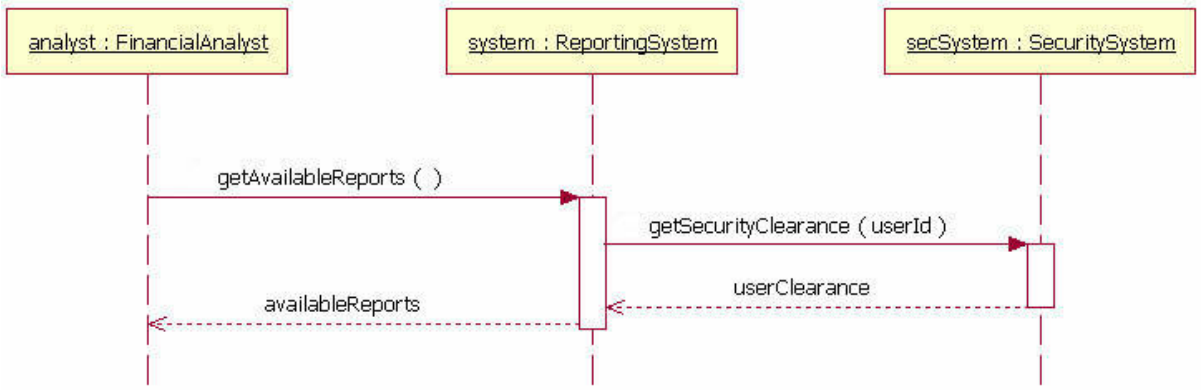
Sequence diagrams

The type of diagrams that describe the interaction between system objects as a SEQUENCE of messages passed between the objects. It describes in what order a group of objects work together.

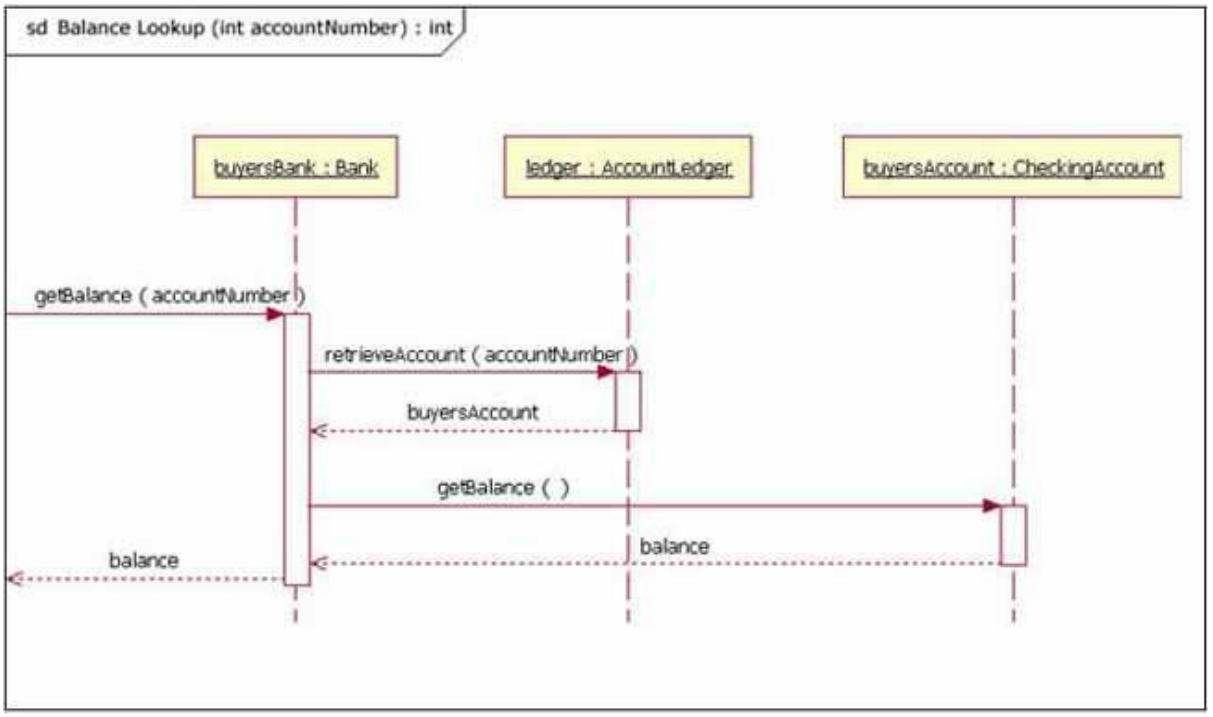
The focus is less on messages themselves and more on the order in which messages occur; nevertheless, most sequence diagrams will communicate what messages are sent between a system's objects as well as the order in which they occur.

One of the primary uses of sequence diagrams is in the transition from requirements expressed as use cases to the next and more formal level of refinement.Use cases are often refined into one or more sequence diagrams.

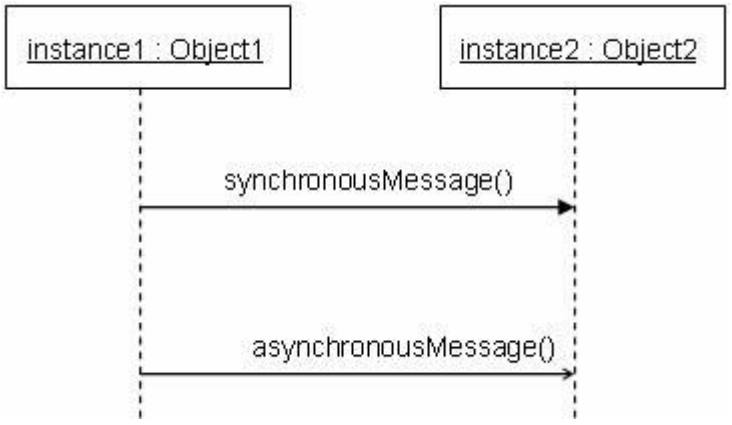
Messages - message that is being sent to the receiving object represents an operation/method that the receiving object's class implements. Sender wants the receiver to do some action



This example shows two synchronous messages (getAvailableReports() and getSecurityClearance()) and two return messages (availableReports and userClearance).

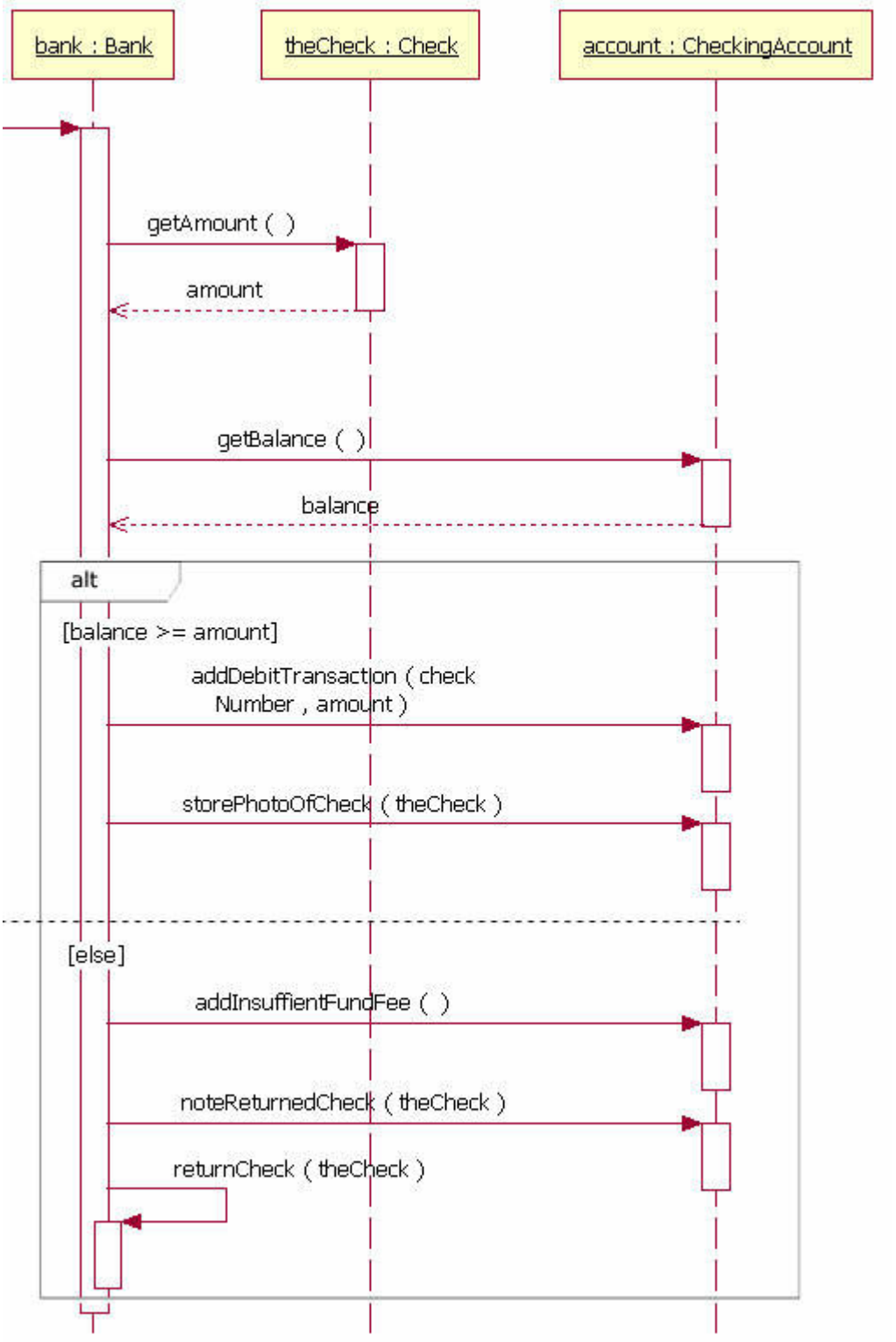


This example shows the balance lookup sequence. First, incoming message getBalance() comes to buyersBank object (instance of Bank class). It is a synchronous message meaning that sender does nothing and waits until buyersBank sends a “return” message back to it. Vertical bars along dashed vertical lines captures how much time does a particular message handling take relative to other messages handling times. During getBalance() message handling, buyersBank sends message ‘retrieveAccount()’ to AccountLedger instance “ledger”, the instance sends “returns” message back to the buyersBank object and so on…

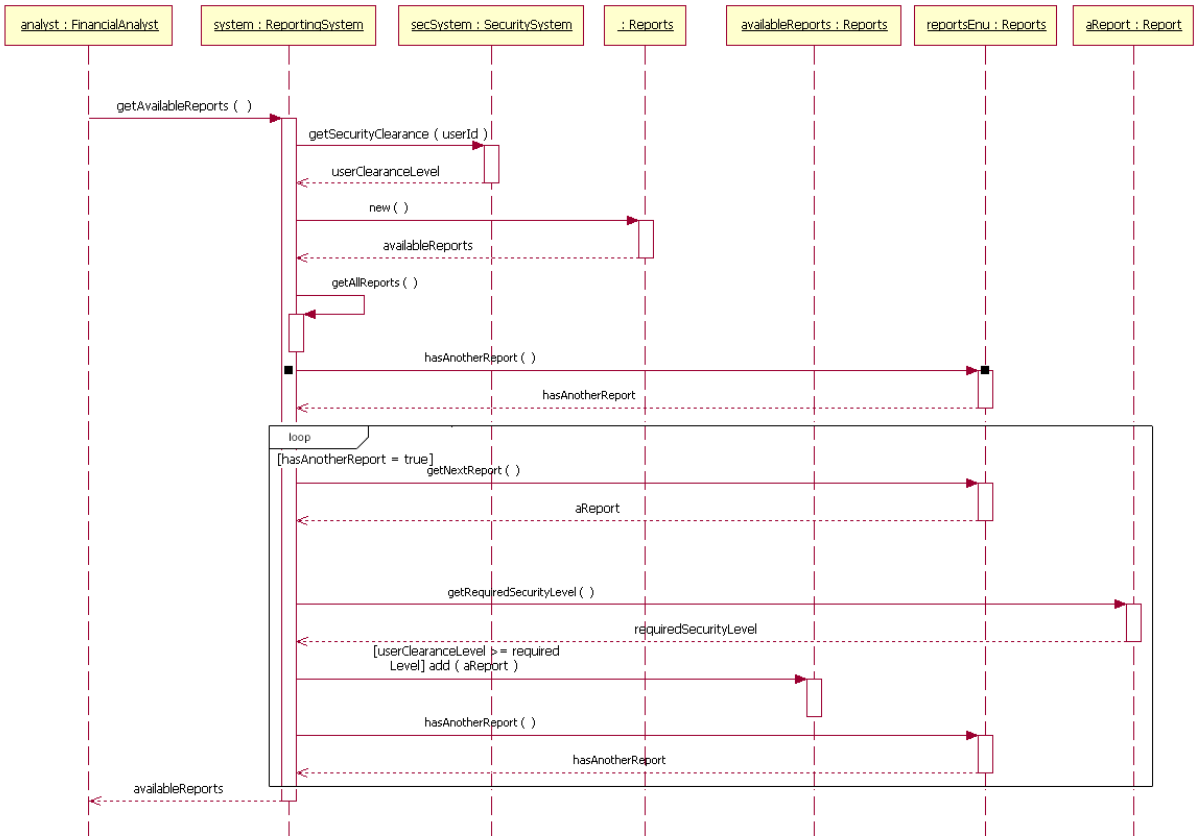


This example show us the difference between sync and async messages

Alternatives - are used to designate a mutually exclusive choice between two or more message sequences. Alternatives allow the modeling of classic “if then else” logic



Loops - are used to repeat subsequences based on some logical value or counter



References

- **https://developer.ibm.com/articles/the-sequence-diagram/**
- https://www.youtube.com/watch?v=WnMQ8HlmeXc
- https://www.youtube.com/watch?v=m8Icp_Cid5o