Topic6: Behavioral Modeling

-Introduction to behavioral modeling
-sequence diagram model
-activity diagram model
-statechart diagram model
-Exercises

6.1 Behavioral modeling

- representing behavioral aspects of any object-oriented system that are constantly changing i.e., either:
  - –interactions between elements through messages,
  - –state transitions and events of elements, or
  - –flow of control among objects and data flows from one activity to another

6.2 Sequence diagram

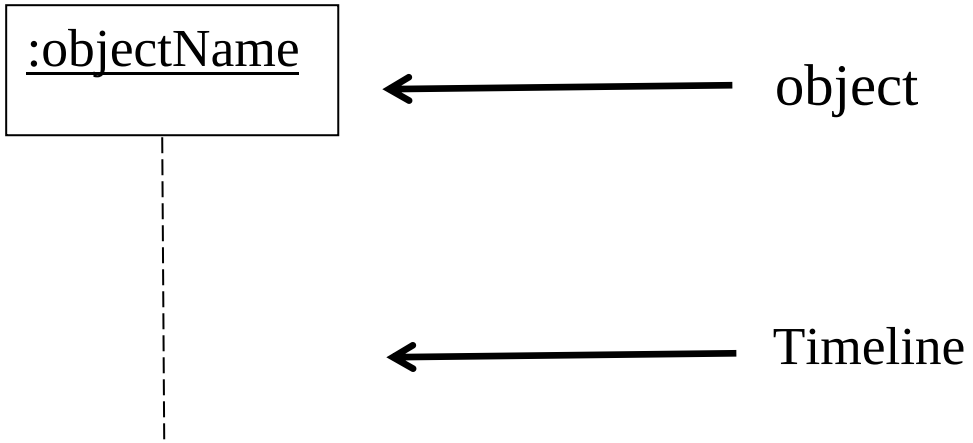
•shows interactions consisting of a set of objects and messages sent and received by objects, with emphasis on the chronological ordering of messages

sequence diagram components

- 1 Object lifeline
- 2 messages
- 3 message syntax

a) Object lifeline & notation

- Object lifeline consists of an object and its timeline that indicates activation and deactivation times of the object



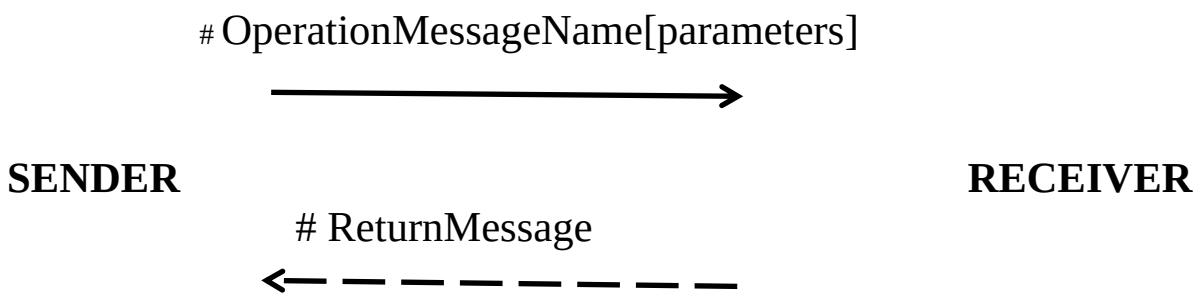
b) Message & notation

•Some type/unit of communication between objects where one may: invoke an operation, raise a signal, cause creation or destruction of target object.

•Modelled as an arrowed line pointing from sender to receiver or viceversa, each message labeled with:

1) named operation invoked in receiver (solid line),

2) or, information returned from receiver (dashed)



c) Message Syntax

predecessors '/' sequence-term iteration [condition] return ':=' operation

where:

1) predecessors is a comma-separated list of sequence numbers of all messages that must come before the current message

2) sequence-term may be either a number or a name that identifies the message

3) iteration determines if a message should be sent once or several times in a sequence: a) one message - add an iteration symbol (*) and a condition to control the number of iterations b) many messages - enclose the set of messages in a box

4) condition specifies the control of the iteration; expressed as a text enclosed within square brackets

5) return may include a list of values sent back to the sender

6) operation defines the name of the operation and optionally its parameters and a return value

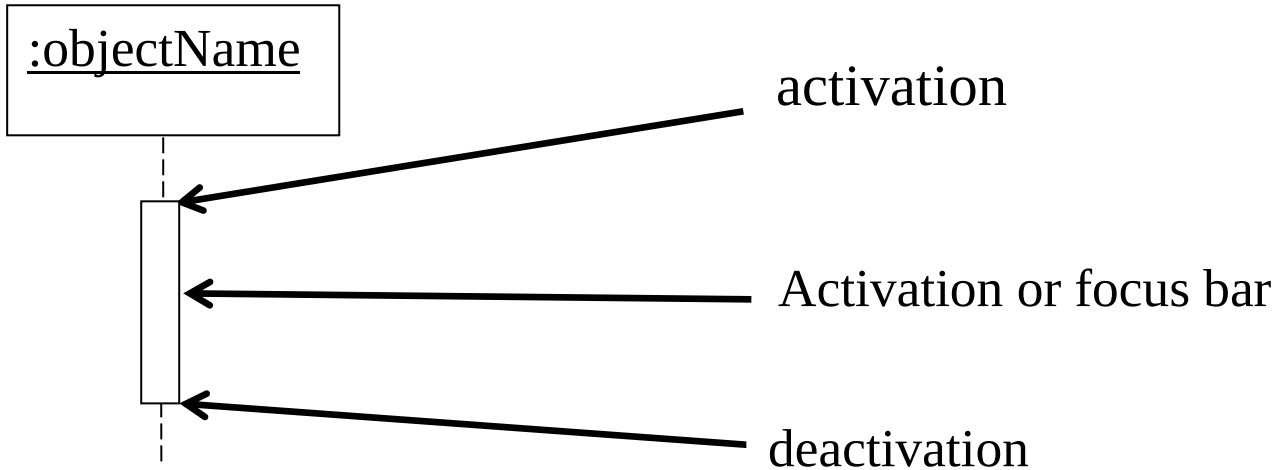
Message Syntax Example

6/8:getAddress * [foreach ApplicationForm]
return text:= getAddress(Citizen.Id:Integer)

1. Specifies message number 8 called getAddress.
2. Message will be executed more than once (*), one time for each ApplicationForm.
3. Each message calls the operation getAddress of the receiving object, sending CitizenId parameter of type Integer, and returns a value of type text.
4. For the execution of this message, it is required that the message 6 has already been executed.

d) Object Activation & Focus of Control

- Object activation is shown by widening the vertical object lifeline to a narrow rectangle, called an activation bar or focus of control.
- An object becomes active at the top of the rectangle and is deactivated when control reaches the bottom of the rectangle.



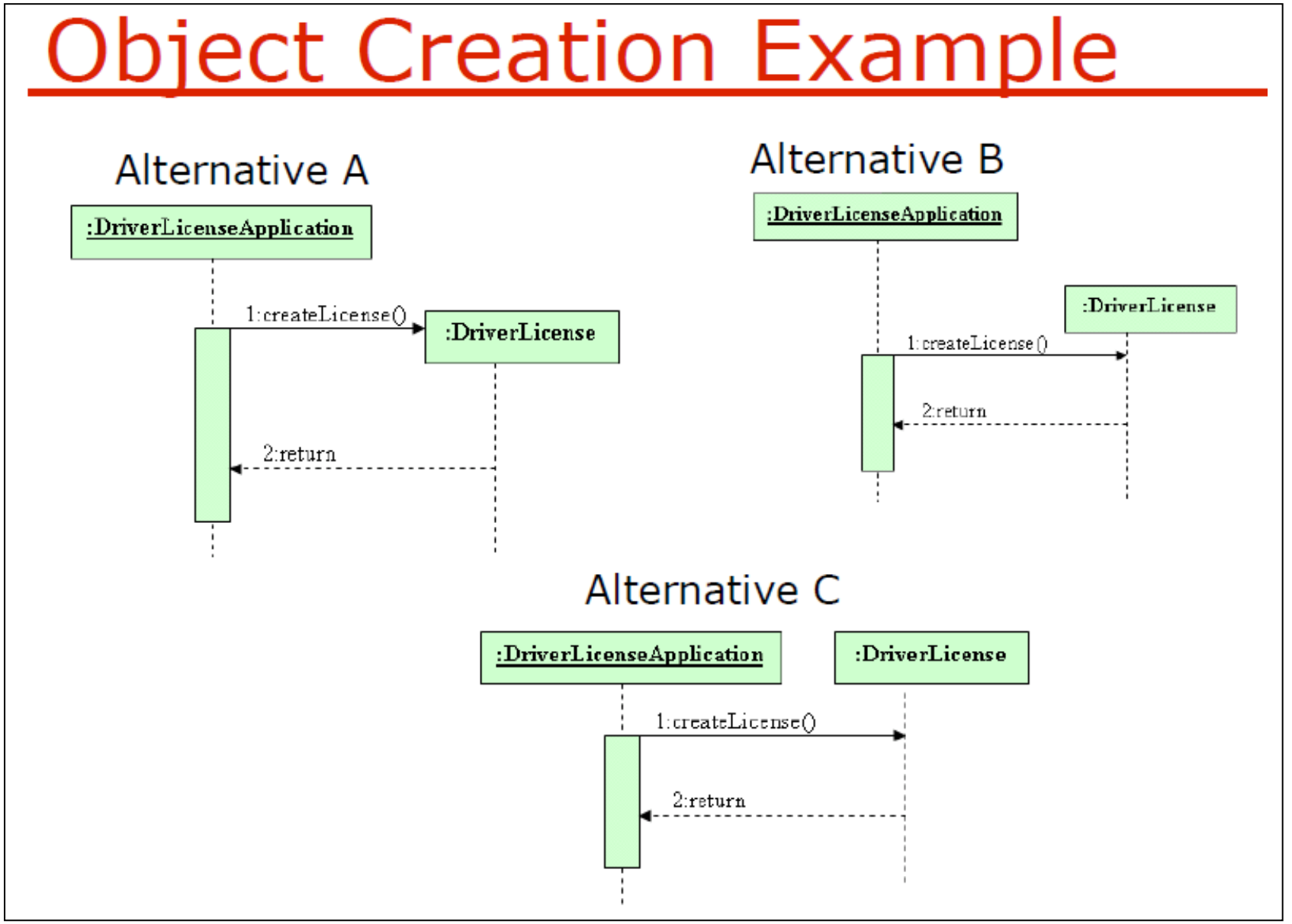
Object Creation,Destruction, Recursion

Object Creation: – if the object is created during the sequence execution it should appear somewhere below the top of the diagram.

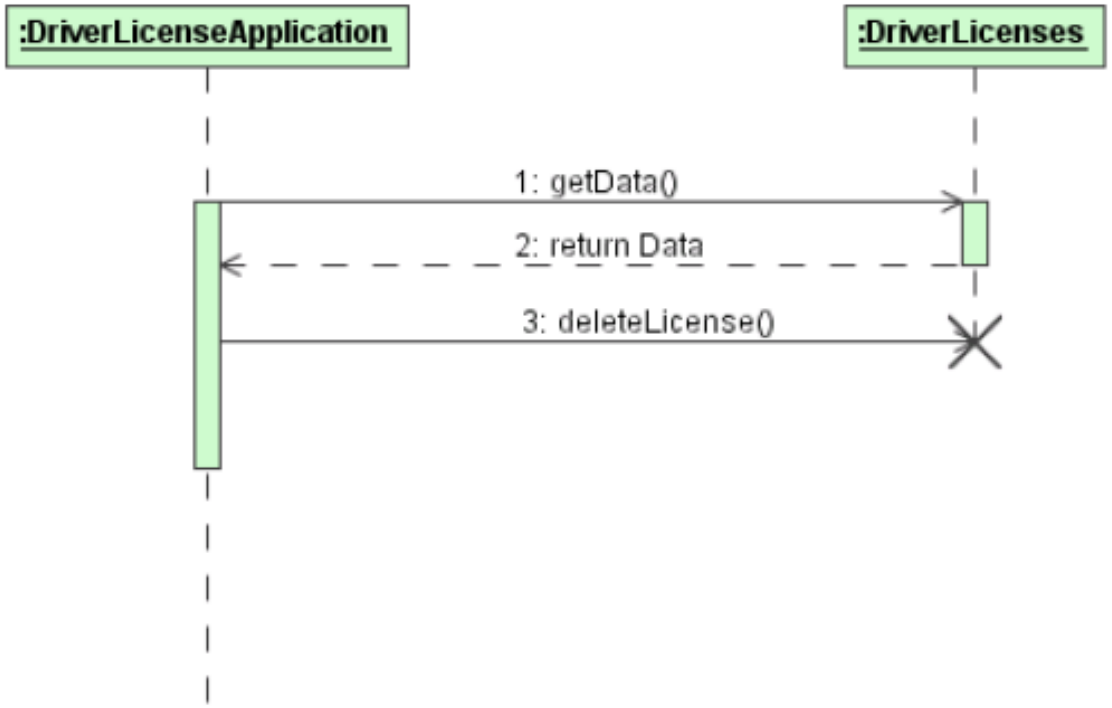
Object Destruction: – if the object is deleted during the sequence execution, place an X at the point in the object lifeline when the termination occurs.

Message Recursion: –An object might also need to call a message recursively; this means to call the same message from within the message.

Object Creation Example



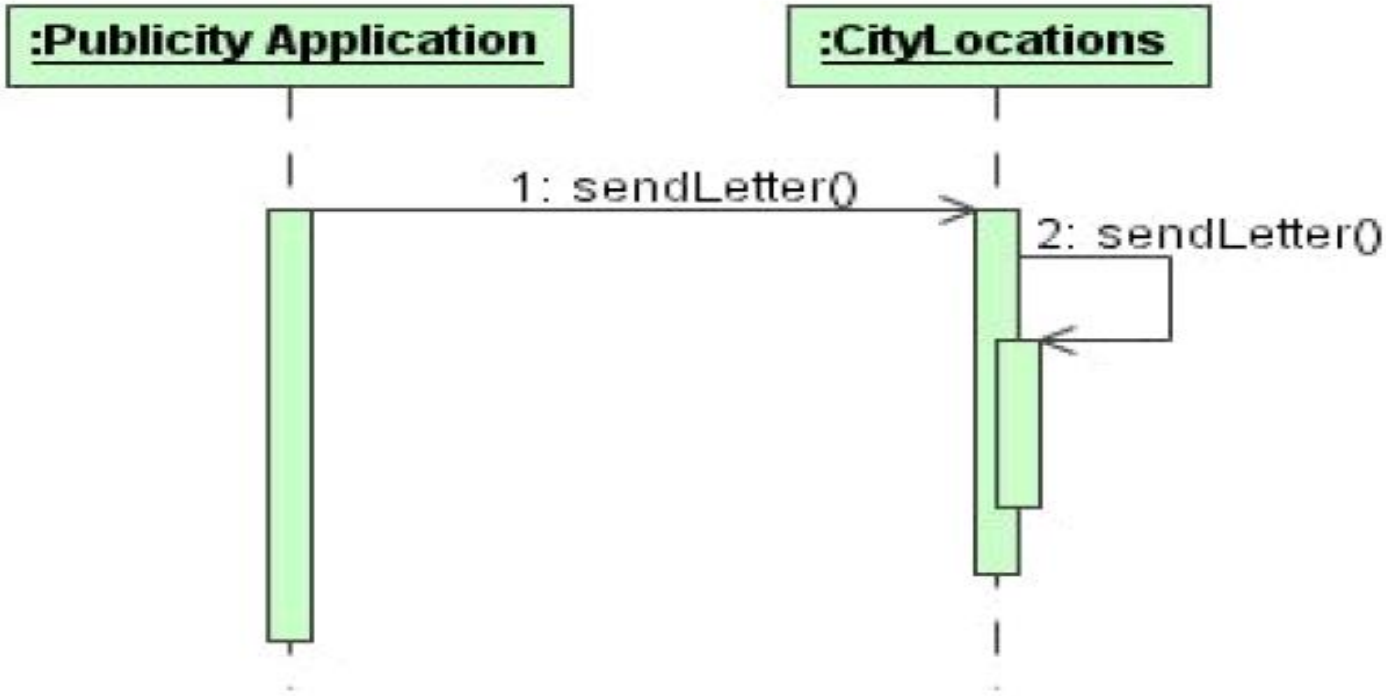
Object Destruction Example



Message Recursion Example

1) suppose that cityLocations is defined in the class diagram as a set of one or more apartments or houses

2) A letter could be sent to all apartments in a location as shown



Example Scenario

An applicant tracks the status of a license application and the system displays the license information.

Procedure:

1. Applicant requests to track the status of a license application
2. System displays the logon form
3. Applicant enters the logon information
4. Applicant submits the logon information
5. System validates the applicant
6. System displays the form to enter the tracking number

7. Applicant enters the tracking number
8. Applicant submits the tracking number
9. System retrieves the license information
10. System displays the license information