

UML. Scenarios and Sequence Diagrams


Software development tools

- Project Risk Assessment
- Project 02: Initial Design
- Project 03: Creating Models
 - ~~Class 20: Midterm Assessment~~
 - ~~Class 21: Model Elements~~
 - Class 22: UML Basics
 - Class 23: Scenarios and Sequence Diagrams
 - Class 24: Consolidating the Model
 - Assessment 07
 - Class 25: Component Diagram Basics
 - Class 26: Diagramming Components
 - Class 27: Class Diagram Basics
 - Assessment 08
 - Class 28: More Class Diagrams
 - Class 29: Diagramming Related Elements
 - Class 30: Classes Vs. Objects
 - Assessment 09

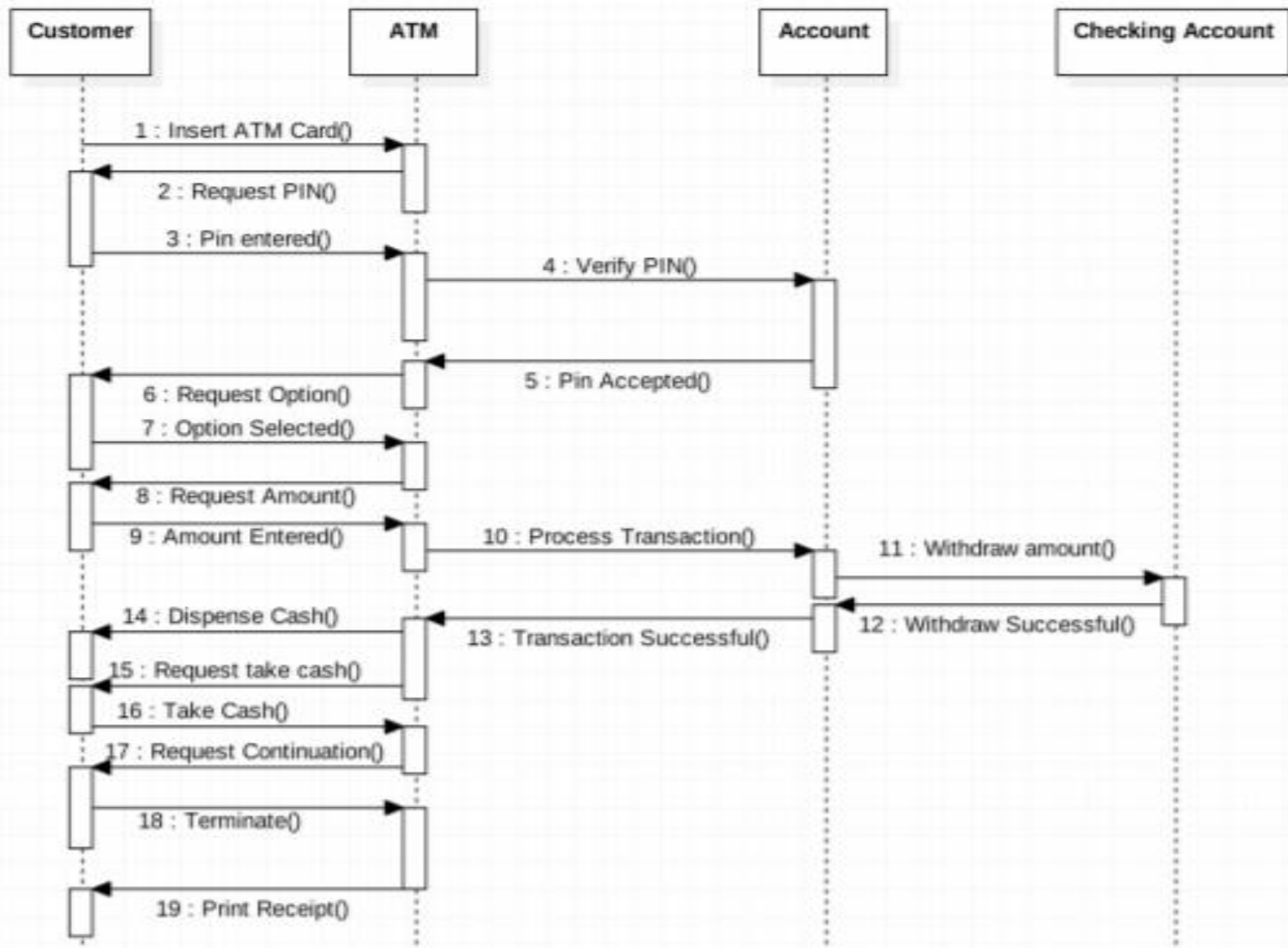
Sequence Diagram

- Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize various runtime scenarios.
- Shows how objects communicate with each other over time.
 - That is, sequence diagrams are used to model object interactions arranged in time sequence and to distribute use case behavior to classes.
 - They can also be used to illustrate all the paths a particular use case can ultimately produce.

Key parts of a sequence diagram

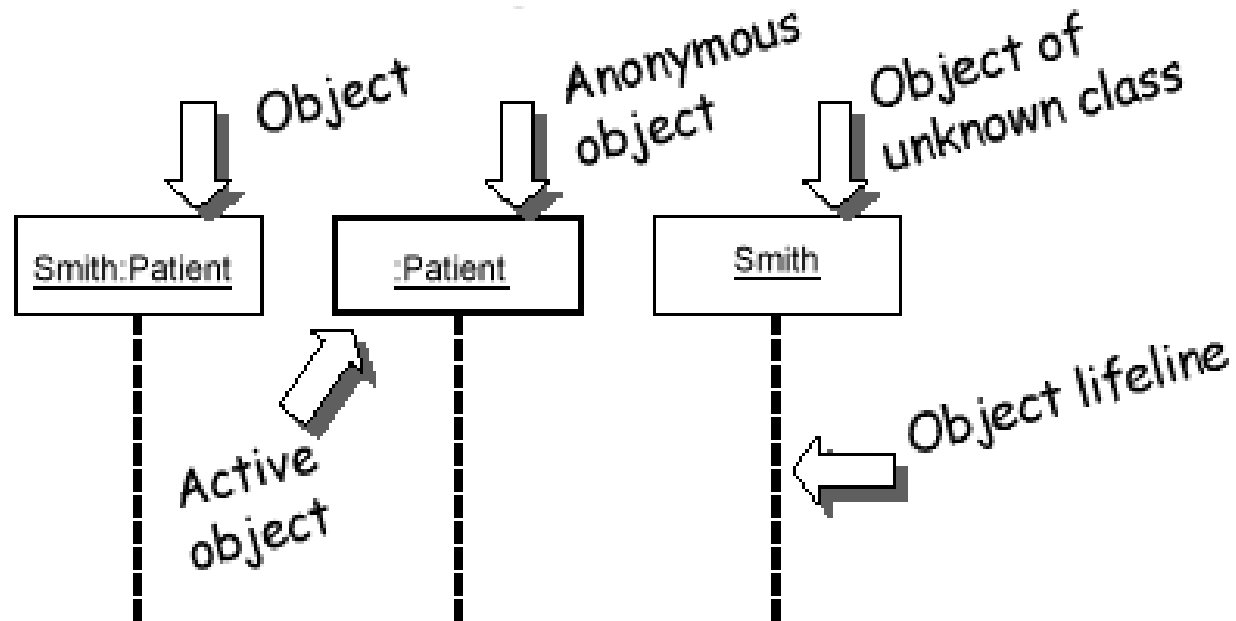
- **participant:** an object or an entity;
the sequence diagram actor
- **message:** communication between objects
- the axes in a sequence diagram:
 - horizontal: which object/participant is acting
 - vertical: time ( forward in time)

Sequence diagram of ATM from use case



Representing objects

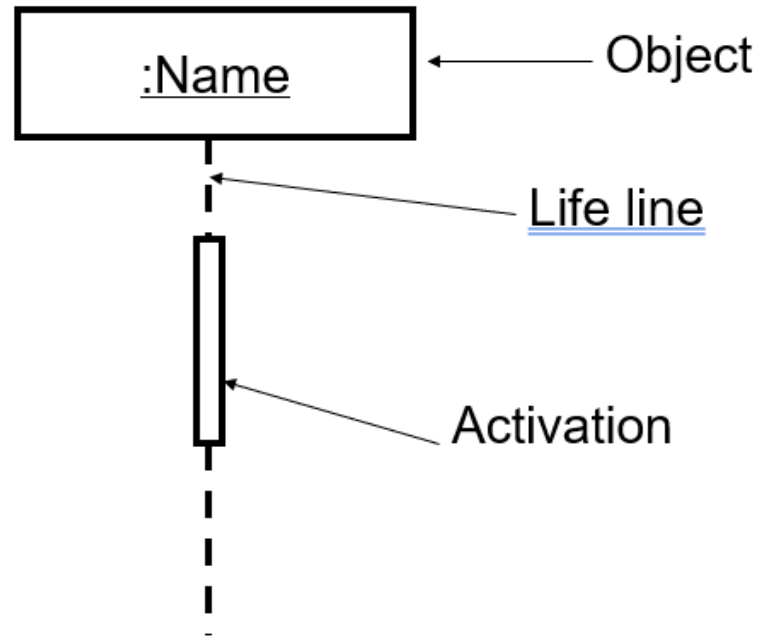
- An object: a square with object type, optionally preceded by object name and colon
 - write object's name if it clarifies the diagram
 - object's "life line" represented by dashed vert. line



Name syntax: <objectname>:<classname>

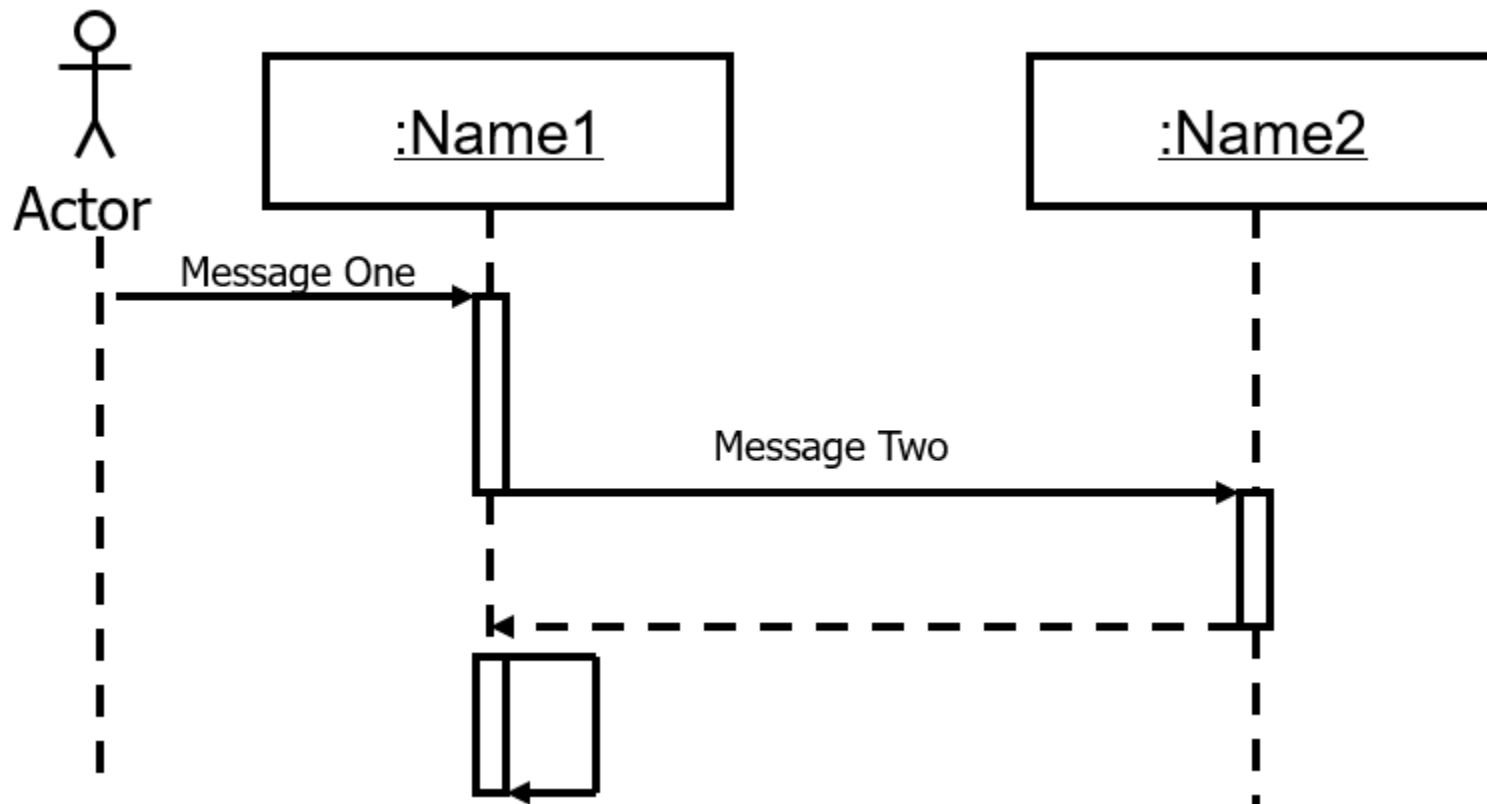
Sequence Diagram - lifeline, Activation

- **Lifelines** are vertical dashed lines that indicate the object's presence over time. **A lifeline** illustrates what is happening to an object in a chronological fashion.
- **activation**: thick box over object's life line.
- **Activation** boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin gray rectangle placed vertically on its lifeline.



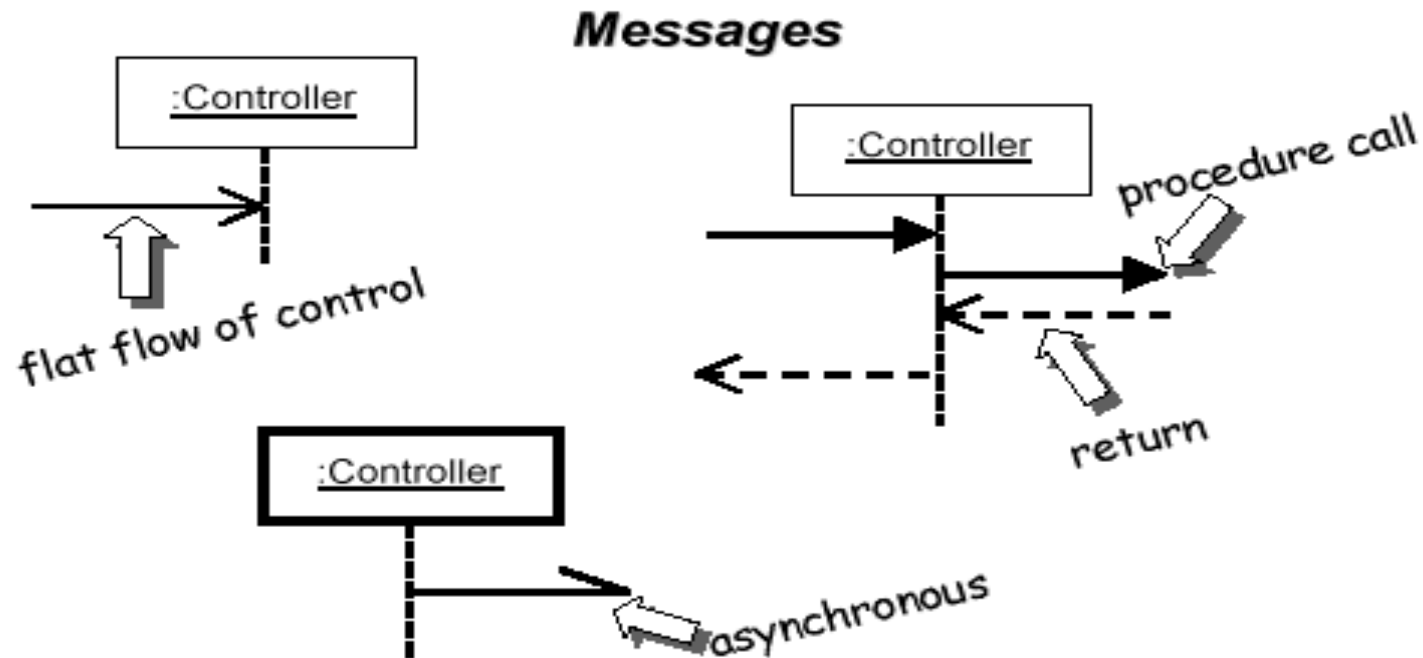
Sequence Diagram – Time & Messages

- Messages are used to illustrate communication between different active objects of a sequence diagram.



Different types of messages

- Type of arrow indicates types of messages
 - dashed arrow back indicates return
 - different arrowheads for normal / asynchronous methods



Types of Messages

- **Synchronous Message**

A synchronous message requires a response before the interaction can continue. It's usually drawn using a line with a solid arrowhead pointing from one object to another.



Synchronous

- **Asynchronous Message**

Asynchronous messages don't need a reply for interaction to continue. Like synchronous messages, they are drawn with an arrow connecting two lifelines; however, the arrowhead is usually open and there's no return message depicted.



Simple, also used for asynchronous



Asynchronous

Types of Messages

- **Reply or Return Message**

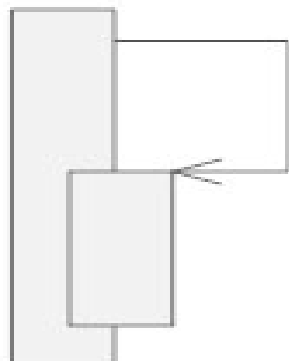
A reply message is drawn with a dotted line and an open arrowhead pointing back to the original lifeline.



Reply or return message

- **Self Message**

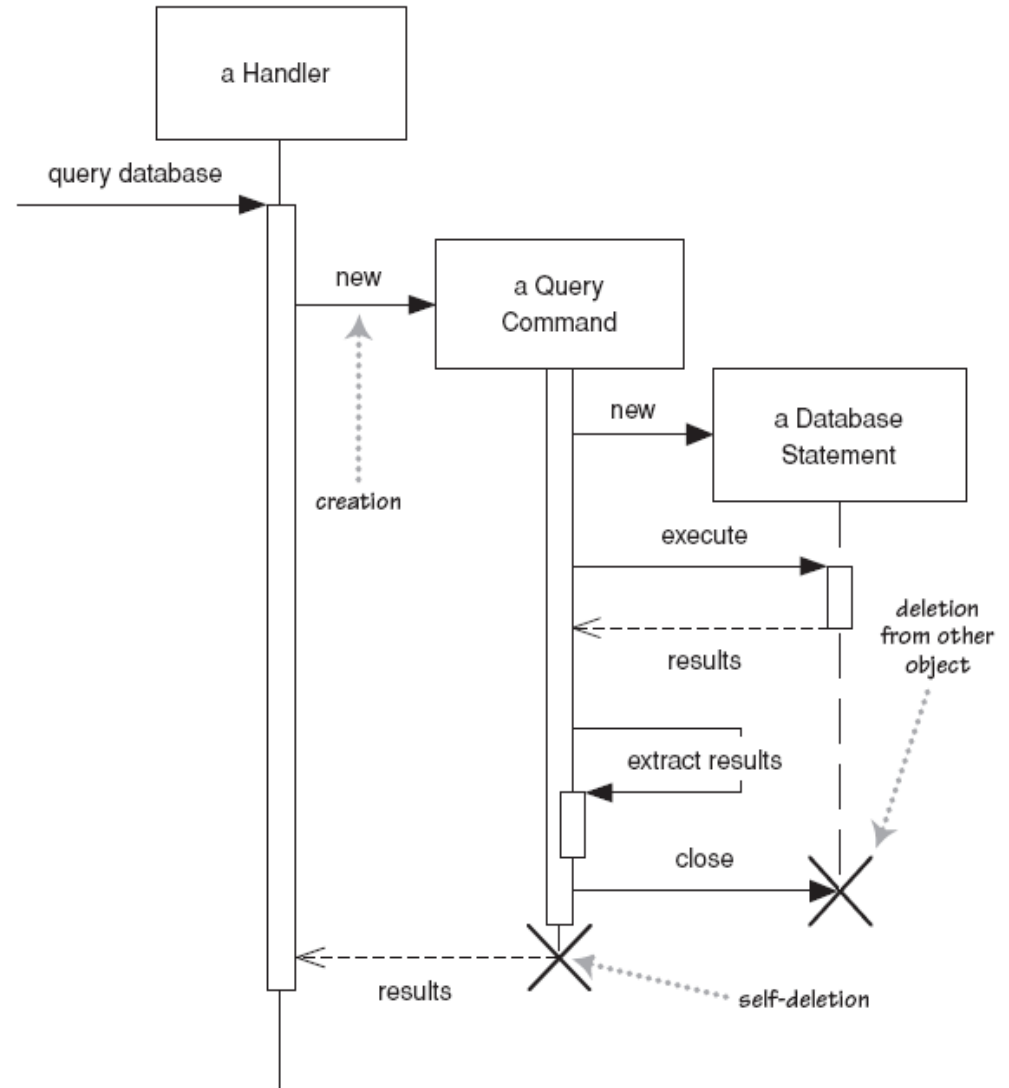
A message an object sends to itself, usually shown as a U shaped arrow pointing back to itself.



Self message

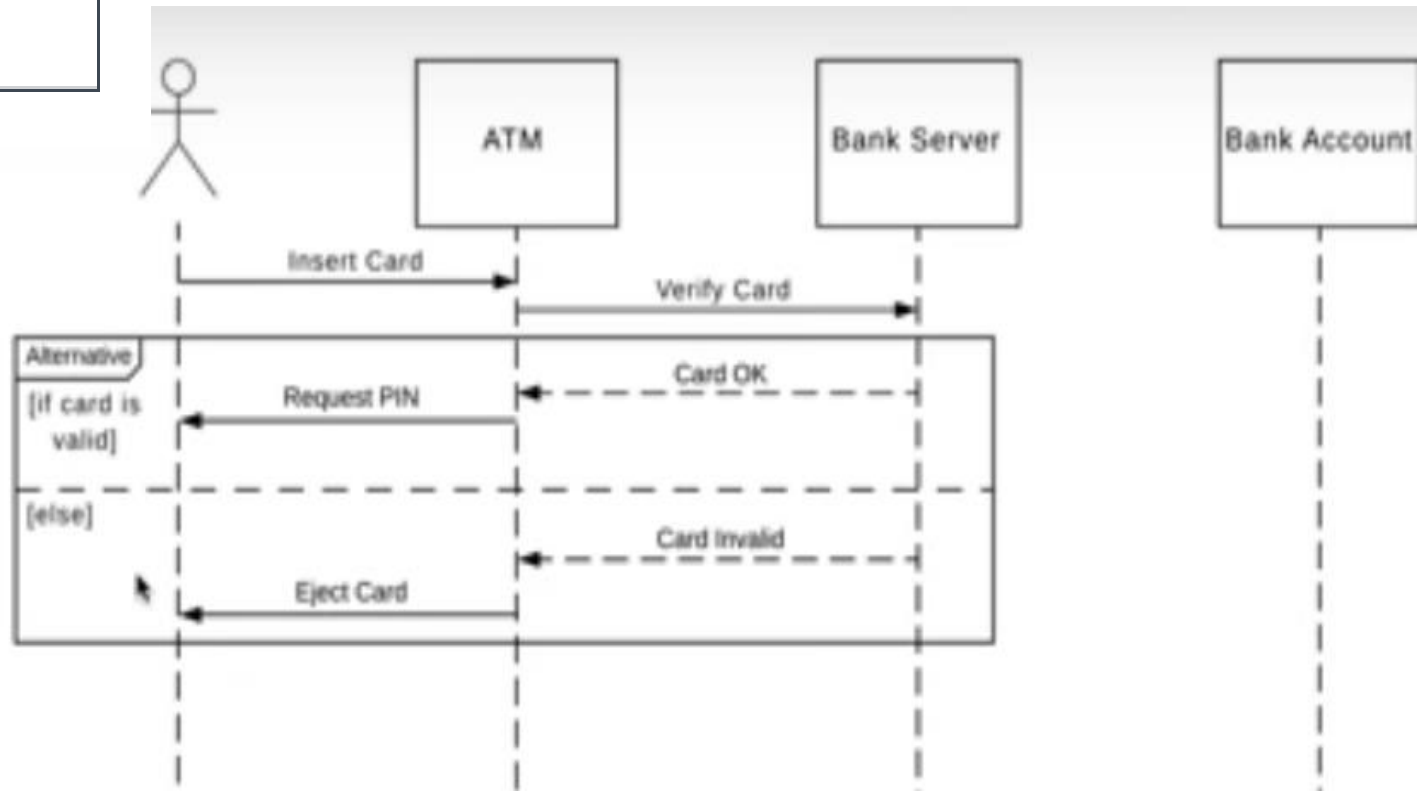
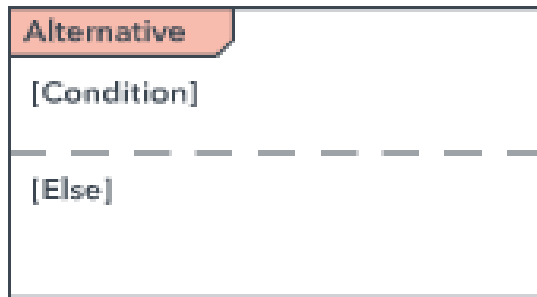
Destroying Objects

- Objects can be terminated.
- This object is removed from memory.
- When that object's lifeline ends, you can place an **X** at the end of its lifeline to denote a destruction occurrence.

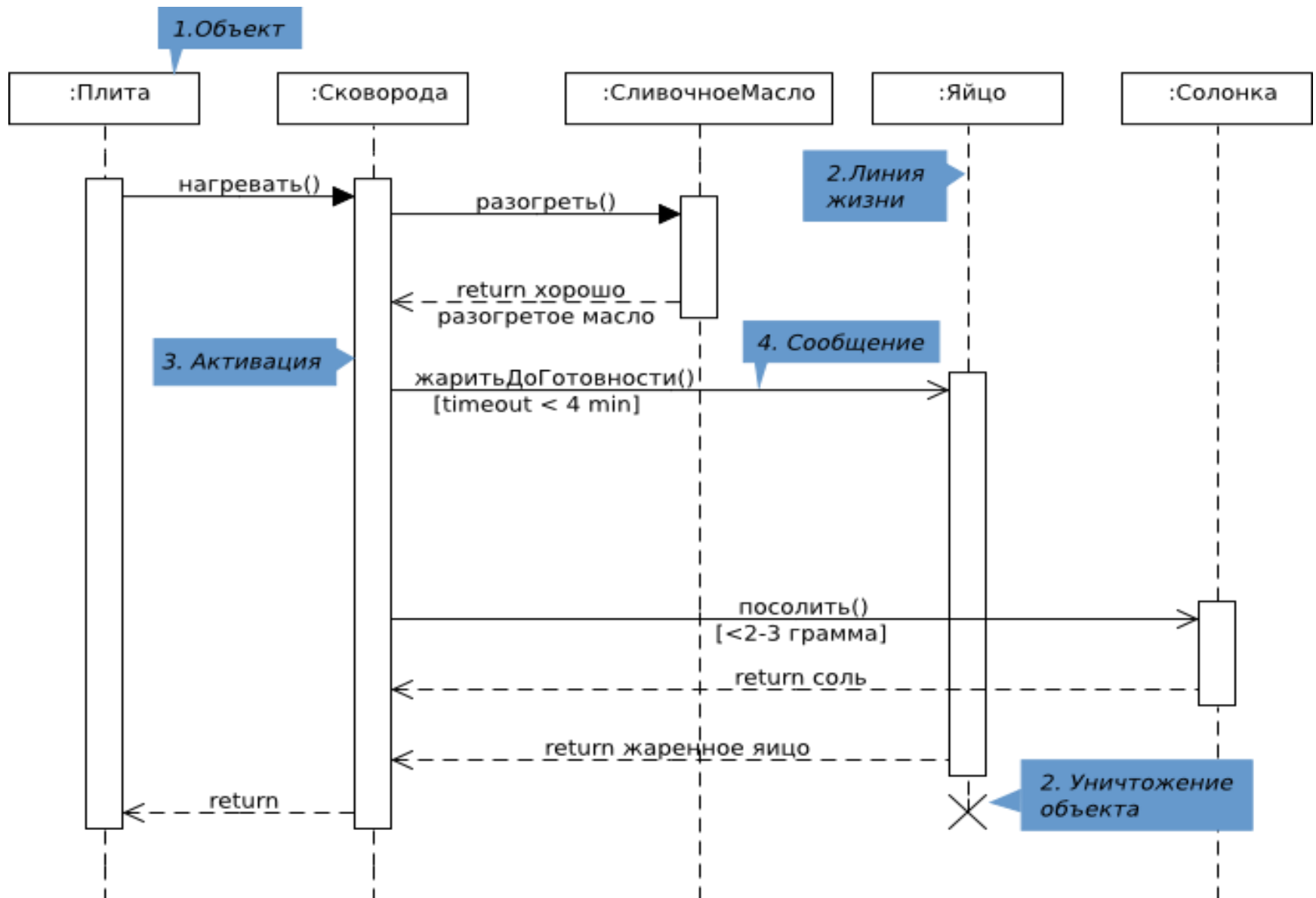


Alternative frame

- Alternative frame symbolizes a choice (that is usually mutually exclusive) between two or more message sequences.



Sequence Diagram Example



ATM Case Study: Sequence Diagram for Withdrawal

